

Escuela Técnica Superior de Ingeniería Universidad de Huelva

Grado en Ingeniería Informática

Trabajo Fin de Grado

Detección y Categorización, según su
Intención, de Mensajes de Contenido
Sexista Mediante Técnicas de Deep
Learning y Aprendizaje con Desacuerdo

Manuel Guerrero García
Junio, 2024

Resumen

El sexismo en redes sociales es un problema prevalente que afecta negativamente a individuos y comunidades, perpetuando estereotipos y comportamientos discriminatorios. Las plataformas sociales, al permitir la libre expresión, a menudo se convierten en espacios donde el lenguaje sexista y misógino prolifera.

Este Trabajo de Fin de Grado se centra en la resolución de la competición EXIST 2024 del CLEF2024, específicamente en las tareas 1 y 2, mediante la aplicación de técnicas de aprendizaje con desacuerdo (*Learning with Disagreement*). La competición EXIST (sEXism Identification in Social neTworks) tiene como objetivo identificar y caracterizar el sexismo, concretamente el Tweets provenientes de la red social Twitter. Las tareas específicas incluyen la clasificación de texto con contenido sexista y la identificación de la intencionalidad del sexismo presente de entre tres clases posibles: sexismo directo, sexismo reportado y sexismo crítico.

Para abordar estos retos, se emplea el paradigma de *Learning with Disagreement*, que se basa en aprovechar las discrepancias en las anotaciones realizadas por diferentes expertos. Este enfoque es crucial para manejar la subjetividad en la anotación de datos y garantizar que los modelos puedan detectar y clasificar el sexismo de manera más robusta y justa.

En este contexto, se utilizan modelos basados en *Transformers*. Los *Transformers* son una tecnología avanzada en el campo del Procesamiento del Lenguaje Natural (PLN) que permite analizar grandes volúmenes de texto y capturar las relaciones contextuales de manera más eficiente que los modelos tradicionales.

El proyecto se estructura en varias etapas, comenzando con una revisión del estado del arte en identificación de sexismo y técnicas de aprendizaje con desacuerdo. Luego, se realiza el diseño experimental, que incluye la selección y preparación de los datos, la configuración de modelos y la implementación de las técnicas propuestas. La fase de experimentación evalúa el rendimiento de los modelos utilizando métricas de evaluación como la F1 e *Information Contrast Measure* (ICM). La métrica F1 es una medida que considera tanto la precisión como la exhaustividad, mientras que ICM evalúa la capacidad del modelo para distinguir entre diferentes clases de datos en presencia de información contrastante.

Finalmente, se analizan los resultados y se presentan conclusiones que destacan tanto las fortalezas como las áreas de mejora de las técnicas utilizadas. Este estudio concluye demostrando el sólido desempeño de los modelos en los que se aplicaron técnicas de aprendizaje con desacuerdo, en comparación con aquellos que no las utilizaron. Estos resultados satisfactorios se reflejan claramente en los rankings finales obtenidos en la competición EXIST 2024, validando así la efectividad del enfoque propuesto.

Palabras clave: *Identificación del Sexismo, Procesamiento del Lenguaje Natural, Transformers, Learning with Disagreement.*

Abstract

Sexism on social media is a prevalent issue that negatively impacts individuals and communities, perpetuating stereotypes and discriminatory behaviors. Social platforms, by allowing free expression, often become spaces where sexist and misogynistic language proliferates.

This Final Degree Project focuses on addressing the EXIST 2024 competition at CLEF2024, specifically on tasks 1 and 2, through the application of Learning with Disagreement techniques. The EXIST competition (sEXism Identification in Social neTworks) aims to identify and characterize sexism, specifically in Tweets from the social network Twitter. The specific tasks include classifying text containing sexist content and identifying the intentionality of the sexism present among three possible classes: direct sexism, reported sexism, and critical sexism.

To tackle these challenges, the Learning with Disagreement paradigm is employed, which leverages discrepancies in annotations made by different experts. This approach is crucial for handling subjectivity in data annotation and ensuring that models can detect and classify sexism more robustly and fairly.

In this context, transformer-based models are used. Transformers are an advanced technology in the field of Natural Language Processing (NLP) that enables the analysis of large volumes of text and captures contextual relationships more efficiently than traditional models.

The project is structured in several stages, beginning with a review of the state of the art in sexism identification and Learning with Disagreement techniques. Then, the experimental design is carried out, which includes data selection and preparation, model configuration, and implementation of the proposed techniques. The experimentation phase evaluates the models' performance using evaluation metrics such as F1 score and Information Contrast Measure (ICM). The F1 score is a metric that considers both precision and recall, while ICM evaluates the model's ability to distinguish between different classes of data in the presence of contrasting information.

Finally, the results are analysed, and conclusions are presented that highlight both the strengths and areas for improvement of the techniques used. This study concludes by demonstrating the solid performance of models applying Learning with Disagreement techniques, compared to those that did not. These satisfactory results are clearly reflected in the final rankings obtained in the EXIST 2024 competition, thereby validating the effectiveness of the proposed approach.

Keywords: *Sexism Identification, Natural Language Processing, Transformers, Learning with Disagreement.*

Agradecimientos

A mis tutores, Jacinto Mata y Victoria Pachón, por su orientación experta, paciencia y dedicación para guiar cada paso de este proyecto hacia su exitosa realización, y por enseñarme tanto en el proceso.

A mis padres, quienes han sido mi mayor apoyo y fuente constante de inspiración a lo largo de este camino académico. Gracias por darme todo lo que he necesitado siempre.

A mi hermano, por llenar mi vida de felicidad y motivarme a ser un ejemplo a seguir para él día tras día.

A mi familia, en especial a mis abuelos, cuyo cariño y aliento han sido fundamentales en mi desarrollo personal, siendo un ejemplo de vida para mí.

A Ali, por su apoyo inquebrantable tanto en lo académico como en lo personal. Agradezco su constante enseñanza y cariño día tras día.

A mis amigos, por su compañía, comprensión y las alegrías compartidas que han hecho esta etapa aún más especial.

Manuel Guerrero García

Huelva, 2024

Índice General

Resumen.....	III
Abstract	IV
Agradecimientos	V
Índice General	VI
Índice de Figuras.....	IX
Índice de Tablas.....	X
1. Introducción.....	1
1.1 Motivación	1
1.2 Objetivos.....	2
1.3 Competencias Adquiridas.....	2
1.4 Estructura de la Memoria.....	3
2. Marco Teórico.....	4
2.1 Aprendizaje Automático.....	4
2.2 Aprendizaje Profundo. Redes Neuronales Profundas.....	5
2.2.1 Estructura de una Red Neuronal	5
2.2.2 Entrenamiento de una Red Neuronal. Conceptos Clave.....	7
2.3 Procesamiento del Lenguaje Natural (PLN)	7
2.4 Transformers.....	8
2.4.1 Arquitectura y Mecanismo de Atención.....	8
2.4.2 Funcionamiento General.....	8
2.4.3 Conceptos Propios de los Transformers	9
2.4.4 Uso de Transformers en este Proyecto	9
2.5 Aprendizaje por Transferencia (<i>Transfer Learning</i>).....	10
2.6 Métricas de Evaluación	10
2.6.1 ICM - <i>Information Contrast Measure</i>	10
2.6.2 Matriz de Confusión	11
2.6.3 Accuracy.....	12
2.6.4 Precision.....	12

2.6.5	Recall.....	13
2.6.6	F1-score.....	13
2.6.7	Curva ROC (<i>Receiver Operating Characteristic</i>).....	13
2.6.8	<i>Area Under the Curve</i> (ROC).....	13
2.7	<i>Ensemble</i> de Modelos	13
2.8	Aprendizaje con Desacuerdo (<i>Learning with Disagreement</i>).....	14
2.9	Tecnologías y Recursos Utilizados	15
3.	Metodología, Experimentación y Resultados.....	16
3.1	Descripción de la Tarea.....	16
3.1.1	Task 1. Identificación de Sexismo.....	17
3.1.2	Task 2. Detección de la Intención del Autor.....	18
3.2	Evaluación de Resultados.....	18
3.3	Descripción de los Datasets Originales.....	19
3.3.1	Composición.....	19
3.3.2	Proceso de Anotación	20
3.3.3	Aplicación de Técnicas de <i>Learning with Disagreement</i>	21
3.4	Descripción General de la Metodología	21
3.5	Selección de los Modelos <i>Transformers</i>	22
3.6	Baseline.....	22
3.6.1	Baseline de la versión A.....	23
3.6.2	Baseline de la versión B.....	24
3.7	Preprocesamiento de Datos.....	26
3.7.1	Descripción de los Datos.....	27
3.7.2	Limpieza y Normalización	29
3.7.3	Balanceo de Datos	30
3.7.4	Tokenización y Codificación	31
3.8	Ajuste de Hiperparámetros	32
3.9	Entrenamiento y Selección de Modelos Finales.....	34
3.9.1	Configuración General de los Entrenamientos.....	34
3.9.2	Identificación de Sexismo en Tweets – Task 1	35
3.9.3	Clasificación de la Intención en Tweets Sexistas – Task 2.....	36
3.10	Validación y Evaluación de los Modelos.....	38
3.10.1	Modelos de la versión v1.1.....	38

3.10.2	Modelos de las versiones v1.3, v1.4 y v1.2.....	39
3.10.3	Modelo de la versión v2.1	39
3.10.4	Modelos de la versión v2.2.....	40
3.10.5	Modelos de la versión v2.3.....	40
3.11	Análisis de Errores	40
3.12	Resultados Oficiales de la Competición.....	41
3.12.1	Ejemplos de Instancias en las <i>Runs</i> Presentadas.....	41
3.12.2	Runs Presentadas para la Task 1	42
3.12.3	Runs Presentadas para la Task 2	43
3.12.4	Resultados Obtenidos	43
4.	Conclusiones y Trabajo Futuro	46
4.1	Conclusiones	46
4.2	Trabajo Futuro	46
4.3	Planificación Temporal del Trabajo Realizado.....	47
	Bibliografía.....	48
	Anexo I. Repositorios de código en GitHub.....	50
	Anexo II. Artículo Científico presentado en EXIST 2024	51

Índice de Figuras

Figura 1 Perceptrón o neurona artificial. (Telefónica, 2020).....	6
Figura 2 “The Transformer - model architecture” (Vaswani, y otros, 2017).....	8
Figura 3 Transfer Learning. Propia.	10
Figura 4 “Confusion Matrix for a binary class dataset” (v7labs, 2022)	11
Figura 5 “Confusion Matrix for a multi-class dataset” (v7labs, 2022)	12
Figura 6 “Técnica de Ensemble de Modelos”	14
Figura 7 “CLEF Editions” (CLEF, 2024)	17
Figura 8 Instancia de los datos contenidos en el fichero “training.json”	20
Figura 9 Método científico aplicado al proyecto. Propia.....	21
Figura 10 Clases posibles en la Task 1 y Task 2. Propia	23
Figura 11 Matriz de confusión para Bert Base M., baseline B	25
Figura 12 Matriz de confusión para Xlm Roberta Base, baseline B.....	25
Figura 13 Matriz de confusión para Deberta v3 Base, baseline B.....	26
Figura 14 Matriz de confusión para Roberta Base Bne, baseline B	26
Figura 15 Datasets empleados para el entrenamiento de los modelos. Propia.....	26
Figura 16 Distribución de clases de la Task 1 y Task 2 en los ficheros training y dev	28
Figura 17 Estudio de Hiperparámetros. Búsqueda: exhaustiva, Modelo: Roberta.....	33
Figura 18 Train output para XLM RoBERTa Base de la versión v1.1	34
Figura 19 Flujo de entrenamiento del modelo v1.1.....	35
Figura 20 Flujo de entrenamiento del modelo v1.2.....	36
Figura 21 Flujo de entrenamiento del modelo v2.1.....	36
Figura 22 Flujo de entrenamiento del modelo v2.2.....	37
Figura 23 Flujo de entrenamiento del modelo v2.3.....	38
Figura 24 Extracto del fichero test proporcionado por la competición	42
Figura 25 Fichero de predicciones (de tipo soft label) presentado a la competición.....	42

Índice de Tablas

Tabla 1 Resultados para el baseline de la versión A	24
Tabla 2 Resultados para el baseline de la versión B en la primera clasificación, binaria	24
Tabla 3 Resultados para el baseline de la versión B en la primera clasificación, multiclasé.....	24
Tabla 4 Comparativa de resultados entre baselines A y B.....	25
Tabla 5 Ejemplos de instancias para la Task 1	27
Tabla 6 Ejemplos de instancias para la Task 2	28
Tabla 7 Limpieza y normalización de datos	30
Tabla 8 Resultados tras el preprocesamiento en los modelos de la Task 1.....	30
Tabla 9 Resultados tras el preprocesamiento en los modelos de la Task 2.....	30
Tabla 10 Ejemplo de generación de datos mediante backtranslation en tweet en español	31
Tabla 11 Ejemplo de generación de datos mediante backtranslation en tweet en inglés	31
Tabla 12 Proceso de tokenizado y codificado de una instancia.....	31
Tabla 13 Espacio de búsqueda de hiperparámetros definido.....	32
Tabla 14 Parámetros seleccionados tras la optimización.....	33
Tabla 15 Resultados para los modelos de la Task 1 tras ajustar hiperparámetros.....	33
Tabla 16 Resultados para los modelos de la Task 2 tras ajustar hiperparámetros.....	33
Tabla 17 Validación y evaluación de los modelos de la versión v1.1	38
Tabla 18 Validación y evaluación de los modelos de la versión v1.3	39
Tabla 19 Validación y evaluación de los modelos de la versión v1.4	39
Tabla 20 Validación y evaluación de los modelos de la versión 1.2	39
Tabla 21 Validación y evaluación de los modelos de la versión 2.2	40
Tabla 22 Validación y evaluación de los modelos de la versión 2.3	40
Tabla 23 Análisis de errores en instancias de la Task 1.....	40
Tabla 24 Análisis de errores en instancias de la Task 2.....	41
Tabla 25 Ranking de participaciones para la evaluación hard-hard en la Task 1	43
Tabla 26 Ranking de participaciones para la evaluación soft-soft en la Task 1	44
Tabla 27 Ranking de participaciones para la evaluación hard-hard en la Task 2	44
Tabla 28 Ranking de participaciones para la evaluación soft-soft en la Task 2	45
Tabla 29 Planificación Temporal del Trabajo Realizado.....	47

CAPÍTULO 1

Introducción

En el entorno digital actual, las redes sociales se han consolidado como plataformas clave para la interacción social y la difusión de ideas. Sin embargo, estas plataformas también se han convertido en vehículos para la propagación de contenidos discriminatorios, incluyendo mensajes de carácter sexista. La prevalencia de este tipo de contenido no solo perpetúa estereotipos de género, sino que también contribuye a un ambiente hostil en línea. La identificación de las intenciones detrás de estos mensajes es crucial para comprender mejor las dinámicas subyacentes que impulsan la discriminación de género en el ámbito digital. Esta comprensión es necesaria no solo para mitigar la difusión de dichos mensajes, sino también para desarrollar estrategias efectivas que promuevan un entorno en línea más seguro y equitativo.

1.1 Motivación

La necesidad de desarrollar herramientas automáticas que puedan identificar y categorizar estos mensajes según su intención es crucial para mitigar su impacto negativo y promover un entorno digital más seguro.

El proyecto fin de carrera se centra en aplicar técnicas avanzadas de procesamiento del lenguaje natural (Jurafsky & Martin, 2024) y aprendizaje profundo, especialmente modelos basados en *Transformers* (Vaswani, y otros, 2017), para abordar esta problemática.

Se propone implementar y adaptar modelos de *Transformers* para analizar tweets y clasificarlos según la posible intencionalidad de su autor. Además de los *Transformers*, este proyecto también se fundamenta en el concepto de *Learning with Disagreement* (Uma, y otros, 2021), una técnica que aborda las diferencias en las anotaciones de datos debido a la subjetividad inherente en la interpretación del lenguaje. El uso de estas tecnologías avanzadas está justificado por varias razones:

- 1. Capacidad de Manejar la Complejidad del Lenguaje Natural:** Los modelos basados en *Transformers* son altamente efectivos en la comprensión y generación de texto, lo que los hace ideales para tareas de clasificación donde el contexto y la sutileza son cruciales.
- 2. Eficiencia en el Procesamiento de Grandes Volúmenes de Datos:** La arquitectura de los *Transformers* permite el procesamiento eficiente de grandes conjuntos de datos, lo cual es esencial para analizar la vasta cantidad de tweets generados diariamente.
- 3. Incorporación de Perspectivas Múltiples:** A través del *Learning with Disagreement*, se pueden capturar diferentes interpretaciones de mensajes potencialmente sexistas, mejorando la robustez y la precisión del modelo.

4. **Mejora Continua Basada en Retroalimentación:** La adaptación y refinamiento constante de los modelos permiten que estos evolucionen y se vuelvan más precisos con el tiempo, basándose en nuevos datos y en la evaluación continua de su rendimiento.

Este enfoque no solo busca identificar la presencia de contenido sexista, sino también comprender la intención detrás de los mensajes, proporcionando una visión más profunda y matizada del problema.

1.2 Objetivos

El objetivo principal de este proyecto es desarrollar modelos de aprendizaje profundo que detecten y categoricen mensajes sexistas en redes sociales, utilizando técnicas avanzadas de procesamiento del lenguaje natural. Para alcanzar este objetivo, primero se realizará una investigación exhaustiva de las técnicas y modelos más avanzados en aprendizaje automático y procesamiento del lenguaje natural aplicados a la clasificación de texto. Posteriormente, se implementarán y adaptarán modelos basados en *Transformers* para analizar y clasificar tweets según su contenido sexista y la intención del autor, integrando técnicas de *Learning with Disagreement* para manejar la subjetividad y variabilidad en la interpretación de estos mensajes. La efectividad de estos modelos será evaluada utilizando métricas de rendimiento adecuadas para asegurar su precisión. Además, se participará en la competición "sEXism Identification in Social neTworks 2024" (Carrillo-de-Albornoz, EXIST: sEXism Identification in Social neTworks, 2024) ofrecida por "Conference and Labs of the Evaluation Forum", (CLEF Initiative, 2016), para validar externamente los resultados y comparar el rendimiento con otros enfoques. El proyecto también buscará mejorar el rendimiento de los modelos pre-entrenados mediante análisis y ajustes basados en los resultados obtenidos. Finalmente, se documentará todo el proceso y los resultados en una memoria detallada, que incluirá la metodología, los modelos implementados, los resultados de la evaluación y las conclusiones. Para ello será necesario aprender y utilizar LaTeX para la redacción y preparación del artículo científico en inglés que documente los hallazgos del proyecto.

1.3 Competencias Adquiridas

Estas competencias han sido seleccionadas de Reglamento específico sobre Trabajo Fin de Grado/Máster de la Escuela Técnica Superior de Ingeniería de la Universidad de Huelva (Ministerio de Educación, Ciencia y Deporte del Gobierno de España, 2018). Las competencias básicas enfatizan la aplicación profesional de conocimientos y el análisis reflexivo de datos relevantes (CB2 y CB3). Las competencias generales incluyen habilidades de análisis y síntesis para integrar información y generar ideas innovadoras (CG0), resolver problemas eficazmente (CG03) y fomentar la innovación (CG09). En cuanto a las competencias específicas, se destaca el dominio y aplicación de sistemas inteligentes (CC15) y el diseño eficiente de estructuras de datos (CC07) para resolver problemas. Además, se subraya la capacidad de evaluar la complejidad computacional y desarrollar soluciones óptimas (CE3-C). Se resalta también el conocimiento profundo de sistemas inteligentes para el diseño y desarrollo de sistemas informáticos (CE4-C), junto con la habilidad para implementar técnicas de

aprendizaje computacional y sistemas de extracción de información en grandes volúmenes de datos (CE7-C) (Ministerio de Educación, Ciencia y Deporte del Gobierno de España, 2018).

1.4 Estructura de la Memoria

Capítulo 1: Introducción. Este capítulo proporciona una visión general del proyecto, estableciendo el contexto y los objetivos de este.

Capítulo 2: Marco Teórico. En este capítulo se exponen los conceptos fundamentales relacionados con el tema de investigación, incluyendo las arquitecturas de modelos y las técnicas de procesamiento de texto empleadas.

Capítulo 3: Metodología, Experimentación y Resultados. Se describe en detalle la metodología utilizada, desde el estudio y preprocesamiento de los datos hasta la implementación y optimización de los modelos de aprendizaje profundo. Se incluye una descripción detallada de la participación en la competición "sEXism Identification in Social neTworks" de 2024. Además, se presentan los resultados obtenidos a partir de la experimentación realizada.

Capítulo 4: Conclusiones y Trabajo Futuro. En este último capítulo, se presentan las conclusiones derivadas del trabajo realizado y se proponen posibles líneas de investigación futuras.

Anexo I: Repositorios de Código en GitHub.

Anexo II: Artículo científico presentado en EXIST 2024.

CAPÍTULO 2**Marco Teórico**

2.1 Aprendizaje Automático

El Aprendizaje Automático, también conocido como Machine Learning (Mitchell, 1997), es una rama de la inteligencia artificial que permite a las computadoras aprender patrones complejos a partir de datos y realizar predicciones o tomar decisiones sin una programación explícita para cada tarea específica. Permite que los ordenadores realicen tareas específicas de manera autónoma, sin la necesidad de una programación previa por parte de un humano. Los algoritmos de aprendizaje automático se dividen en tres categorías principales, siendo las dos primeras las más utilizadas:

- **Aprendizaje Supervisado:** En este enfoque, el algoritmo se entrena utilizando un conjunto de datos etiquetados, es decir, con resultados esperados conocidos. Esto permite crear un modelo capaz de predecir una salida basada en una entrada diferente a la del conjunto de entrenamiento (Mitchell, 1997). Por ejemplo, en la clasificación de textos, un caso de uso específico podría ser la identificación de sentimientos en reseñas de productos, donde el modelo se entrena con reseñas etiquetadas como positivas o negativas.
- **Aprendizaje no Supervisado:** Este tipo de aprendizaje no requiere conocimiento previo de los resultados esperados y busca patrones en un conjunto de datos desordenados (Hastie, Tibshirani, & Friedman, 2009). En el contexto de la clasificación de textos, el aprendizaje no supervisado podría ser útil para agrupar textos similares sin etiquetas previas, como organizar noticias en diferentes categorías temáticas basadas en su contenido semántico.
- **Aprendizaje por Refuerzo:** En este enfoque, el modelo interactúa con un entorno y aprende a través de la retroalimentación proporcionada por las acciones que realiza (Sutton & Barto). Utilizando un sistema de recompensas, el modelo aprende a realizar acciones que maximicen la recompensa acumulada a lo largo del tiempo. Aunque no se menciona en el ejemplo proporcionado, este tipo de aprendizaje podría aplicarse en la clasificación de textos para mejorar la precisión del modelo a través de la retroalimentación proporcionada por usuarios o expertos.

En el contexto de este trabajo fin de grado, se ha utilizado Aprendizaje Supervisado para resolver la tarea de clasificación de textos, ya que se cuenta con un conjunto de datos etiquetados para entrenar el modelo y predecir la clase de nuevos textos.

2.2 Aprendizaje Profundo. Redes Neuronales Profundas

El Aprendizaje Profundo, o Deep Learning, es una subdisciplina del aprendizaje automático que utiliza redes neuronales artificiales con múltiples capas para procesar y aprender de grandes volúmenes de datos (Goodfellow, Bengio, & Courville, 2016). Estas redes neuronales profundas están diseñadas para imitar el funcionamiento del cerebro humano, permitiendo la extracción de características y la creación de representaciones complejas de los datos.

Las redes neuronales profundas consisten en múltiples capas de neuronas artificiales, donde cada capa procesa los datos de manera que se construyen representaciones cada vez más abstractas (Schmidhuber, 2015).

2.2.1 Estructura de una Red Neuronal

Las redes neuronales artificiales están inspiradas en la estructura y funcionamiento del cerebro humano. En su esencia, una red neuronal está formada por unidades básicas de procesamiento llamadas neuronas artificiales. Cada neurona artificial recibe entradas, que pueden ser características de los datos de entrada, y aplica pesos a esas entradas para determinar su importancia relativa. Además, se agrega un sesgo, un valor adicional que ayuda a ajustar la salida de la neurona. Estas entradas ponderadas se pasan a través de una función de activación, que transforma la suma ponderada en la salida de la neurona. Durante el entrenamiento de la red neuronal, esta estructura se utiliza repetidamente para procesar datos y realizar predicciones. Luego, estas predicciones se comparan con los valores reales para calcular el error. Utilizando este error, se ajustan los pesos y sesgos de las neuronas en un proceso conocido como retropropagación (*backpropagation*). Este proceso iterativo permite que la red neuronal aprenda y mejore su rendimiento en la tarea para la que fue diseñada, ajustando continuamente sus parámetros para minimizar el error entre las predicciones y los valores reales. La fórmula que define el funcionamiento de una neurona artificial en una red neuronal es la siguiente:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

- **Entradas** (x_1, x_2, \dots, x_n): Valores que la neurona recibe, que pueden ser características de los datos de entrada.
- **Pesos** (w_1, w_2, \dots, w_n): Parámetros que se ajustan durante el entrenamiento para determinar la importancia de cada entrada.
- **Bias (Sesgo)** (b): Un valor adicional que se suma a la ponderación de las entradas, ayudando a ajustar la función de activación.
- **Función de Activación** (f): Transforma la suma ponderada de las entradas más el bias (sesgo) en la salida de la neurona.

El Perceptrón Multicapa. Capas

El Perceptrón Multicapa (Grosse, 2018) es una arquitectura de red neuronal artificial (RNA) que consta de múltiples capas de neuronas interconectadas. En comparación con el concepto anterior de una sola neurona, el Perceptrón Multicapa extiende esta idea al organizar las neuronas en capas.

- **Capa de Entrada:** Es la primera capa de la red y recibe los datos de entrada. No realiza cálculos, simplemente pasa los valores a la siguiente capa.
- **Capas Ocultas:** Son capas intermedias entre la capa de entrada y la capa de salida. Aquí es donde se lleva a cabo la mayor parte del procesamiento de la información. Cada neurona en una capa está conectada a todas las neuronas de la siguiente capa, lo que se conoce como redes completamente conectadas.
- **Capa de Salida:** Es la última capa de la red y proporciona la salida final. La naturaleza de esta capa varía según el tipo de problema a resolver, como regresión o clasificación.

Esta estructura multicapa permite que la red neuronal aprenda y modele relaciones más complejas entre los datos de entrada y las salidas deseadas, lo que la hace más poderosa y versátil en la resolución de una amplia gama de problemas de aprendizaje automático. En la Figura 1 se presenta una esquematización del perceptrón.

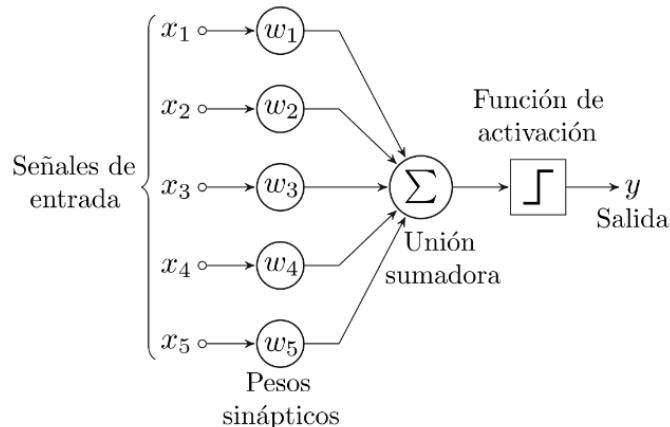


Figura 1 Perceptrón o neurona artificial. (Telefónica, 2020)

Funciones de Activación

Su principal propósito es introducir no linealidad en el modelo, permitiendo que la red neuronal aprenda y represente relaciones complejas en los datos. Algunas de estas son las siguientes:

- **Sigmoide.** $\sigma(x) = \frac{1}{1+e^{-x}}$
- **Tanh.** $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
- **ReLU (Rectified Linear Unit).** $f(x) = \max(0, x)$
- **Softmax.** Es usada en la capa de salida para problemas de clasificación multiclase. $\sigma(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$

2.2.2 Entrenamiento de una Red Neuronal. Conceptos Clave

El entrenamiento de una red neuronal implica una serie de procesos interrelacionados que son fundamentales para su aprendizaje efectivo. La inicialización de pesos establece los parámetros iniciales de la red neuronal, lo cual es crucial para evitar problemas como la desaparición o explosión del gradiente durante el entrenamiento. Durante la propagación hacia adelante (*forward propagation*), los datos de entrada atraviesan cada capa de la red neuronal, donde se aplican funciones de activación para calcular las salidas de cada neurona hasta obtener la salida final de la red. La función de pérdida evalúa la discrepancia entre las predicciones de la red neuronal y los valores reales esperados. Utiliza diferentes métricas según el tipo de problema, como la entropía cruzada para clasificación o el error cuadrático medio para regresión (Goodfellow I. B., 2016). El *backpropagation* calcula los gradientes de la función de pérdida respecto a cada peso en la red. Este proceso utiliza la regla de la cadena para propagar el error hacia atrás a través de las capas de la red, permitiendo ajustar los pesos para minimizar la pérdida. La optimización actualiza los pesos de la red utilizando algoritmos como el descenso de gradiente estocástico (SGD). Estos algoritmos ajustan el *learning rate* de manera adaptativa para mejorar la eficiencia del entrenamiento. Para prevenir el *sobreajuste (overfitting)*, se utilizan técnicas de regularización como *Dropout*, que desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para evitar dependencias complejas entre neuronas, *Weight Decay*, que añade un término de penalización a la función de pérdida para penalizar pesos grandes, y *Early Stopping*, que detiene el entrenamiento cuando la pérdida en el conjunto de validación deja de mejorar. El entrenamiento en *minibatches* implica utilizar pequeñas muestras de datos en lugar de todo el conjunto de datos para cada actualización de los pesos. Esto hace que el entrenamiento sea más eficiente y estable, ya que reduce la variabilidad en la estimación del gradiente y permite utilizar más datos en el proceso de entrenamiento.

2.3 Procesamiento del Lenguaje Natural (PLN)

El Procesamiento del Lenguaje Natural (PLN) es una disciplina de la inteligencia artificial que se centra en la interacción entre las computadoras y el lenguaje humano (Jurafsky & Martin, 2024). Su objetivo es permitir a las máquinas comprender, interpretar y generar texto en lenguaje humano de manera similar a como lo hacen los humanos. En el PLN se emplean una variedad de técnicas y algoritmos para realizar diversas tareas, como el análisis de sentimientos, la traducción automática, la extracción de información y la generación de resúmenes, entre otras. Estas tareas pueden involucrar el procesamiento de texto en diferentes niveles, desde el nivel fonético y morfológico hasta el nivel semántico y pragmático. Por ejemplo, mediante el uso de algoritmos de PLN, es posible realizar tareas como la tokenización, que divide el texto en palabras o frases significativas; la lematización, que reduce las palabras a su forma base; y el análisis de sentimientos, que determina la intencionalidad detrás de las palabras.

2.4 Transformers

Los *Transformers* son una arquitectura de redes neuronales que ha revolucionado el procesamiento del lenguaje natural. Utilizan un mecanismo de atención para capturar relaciones de dependencia a largo plazo entre palabras, lo que los hace altamente escalables y efectivos en tareas de PLN.

2.4.1 Arquitectura y Mecanismo de Atención

La arquitectura de un *Transformer* incluye un codificador y un decodificador, cada uno compuesto por múltiples capas de atención y *feed-forward*. La capa de atención es fundamental, ya que permite al modelo enfocarse en partes específicas de la secuencia de entrada durante la codificación o decodificación. En el mecanismo de atención, se calcula una puntuación para cada par de palabras en la secuencia, que se utiliza para ponderar las representaciones de palabras y concentrar la atención en las más relevantes para la tarea. Este proceso se repite en varias capas de atención, permitiendo al *Transformer* capturar relaciones de dependencia a diferentes niveles de abstracción en los datos de entrada.

2.4.2 Funcionamiento General

Su arquitectura se distingue por emplear mecanismos de atención que capturan dependencias a largo plazo en los datos. Esta capacidad se potencia con codificadores y decodificadores en capas, lo que facilita un aprendizaje jerárquico profundo de las representaciones de secuencias. En detalle, el proceso comienza transformando la secuencia de entrada en embeddings, añadiendo información posicional para preservar el orden de las palabras. Luego, el codificador procesa esta secuencia a través de múltiples capas de atención y perceptrones multicapa, con normalizaciones para estabilizar el aprendizaje. La salida del codificador se transfiere al decodificador, que comienza transformando la secuencia objetivo en embeddings similares, también con información posicional. Durante la decodificación, el *Transformer* utiliza atención enmascarada para enfocarse solo en partes relevantes de la secuencia objetivo. Posteriormente, atiende la secuencia codificada para capturar conexiones importantes entre ambas secuencias. Véase esto en la Figura 2.

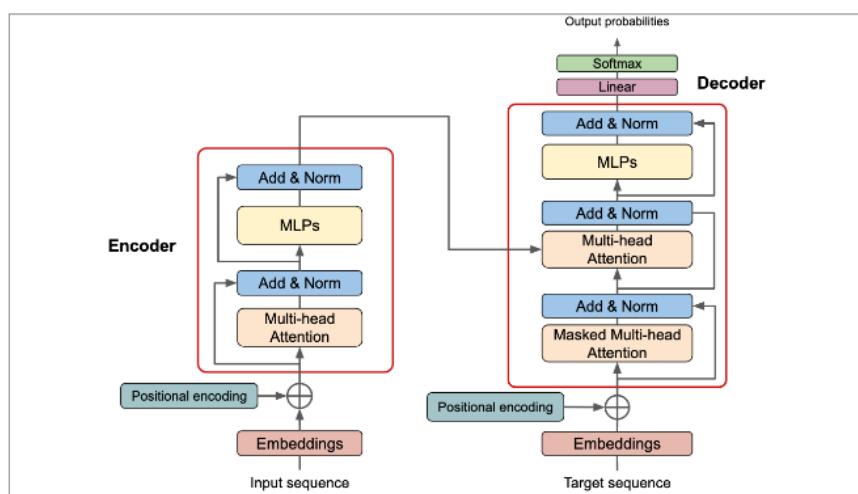


Figura 2 “The Transformer - model architecture” (Vaswani, y otros, 2017)

Este proceso incluye capas de perceptrones y normalizaciones, refinando las representaciones en cada paso. Finalmente, la salida del decodificador se procesa mediante una capa lineal y softmax, generando probabilidades sobre el vocabulario y produciendo la salida final de manera precisa y contextualmente informada.

2.4.3 Conceptos Propios de los Transformers

Dentro del contexto de los *Transformers*, se destacan los siguientes conceptos clave:

- **Longitud Máxima (Max Length):** Es crucial para controlar la complejidad computacional y el consumo de recursos. Limita la longitud de las secuencias de entrada para evitar tiempos de entrenamiento prolongados y altos requisitos de memoria.
- **Padding:** Es esencial para igualar la longitud de todas las secuencias de entrada en un lote de datos, permitiendo el procesamiento eficiente en paralelo.
- **Embeddings:** Transforman palabras o *tokens* en vectores numéricos, capturando la semántica y el contexto de manera más efectiva que las representaciones discretas tradicionales.
- **Softmax:** Utilizada en dos aspectos principales: en el mecanismo de atención para calcular ponderaciones y en la salida final para convertir *logits* en probabilidades, lo que facilita la generación de secuencias y la clasificación de palabras.
- **Mecanismo de Atención:** Es una de las características distintivas de los *Transformers*. Permite que el modelo determine la importancia relativa de diferentes partes de una secuencia en relación con otras partes. Esto se logra calculando ponderaciones de atención para cada par de palabras en la secuencia de entrada, lo que permite al modelo enfocarse en las partes relevantes del contexto al realizar tareas como la traducción automática o la generación de texto.

2.4.4 Uso de Transformers en este Proyecto

El desarrollo de este proyecto implica utilizar modelos basados en *Transformers* debido a su notable capacidad para la clasificación de textos. Los *Transformers* destacan por su habilidad para capturar relaciones contextuales entre palabras, lo cual es crucial para mejorar la precisión de las predicciones en el análisis de lenguaje natural. Estos modelos aprenden representaciones contextualizadas de palabras, permitiendo una comprensión más profunda y precisa del significado de las palabras según su contexto en la oración. Por ejemplo, en un tweet que contenga lenguaje ambiguo, los *Transformers* pueden discernir la intención detrás de las palabras basándose en su entorno textual, mejorando así la detección de contenido sexista y su intencionalidad. Además, los *Transformers* pre-entrenados ofrecen una ventaja significativa ya que pueden adaptarse a tareas específicas de clasificación de textos mediante el ajuste fino. Esto significa que se puede aprovechar el conocimiento previo adquirido durante el preentrenamiento, sin necesidad de iniciar el entrenamiento desde cero, lo que ahorra tiempo y recursos.

2.5 Aprendizaje por Transferencia (*Transfer Learning*)

El aprendizaje por transferencia es una técnica en el campo del aprendizaje automático que consiste en transferir conocimientos adquiridos en una tarea a otra tarea relacionada (Pan & Yang, 2010). En lugar de entrenar un modelo desde cero para cada tarea específica, se aprovecha el conocimiento aprendido de problemas previos para mejorar el rendimiento en una nueva tarea. El aprendizaje por transferencia en aprendizaje automático implica aprovechar conocimientos previos de tareas relacionadas para mejorar el rendimiento en nuevas tareas, en lugar de entrenar modelos desde cero. Esto es especialmente beneficioso con conjuntos de datos limitados o cuando el entrenamiento desde cero es costoso en recursos y tiempo computacionales. El ajuste fino es una aplicación específica de este enfoque, donde un modelo pre-entrenado se adapta a una tarea específica modificando sus pesos durante el entrenamiento adicional. Esto permite una adaptación más precisa y eficiente del modelo, como en el caso del modelo RoBERTa Base BNE (Rodríguez, 2021), que inicialmente fue pre-entrenado en un corpus extenso de texto en español. Para aplicaciones como la clasificación de texto, el ajuste fino en conjuntos de datos específicos mejora significativamente el rendimiento del modelo al transferir y adaptar el conocimiento previamente adquirido. En la Figura 3, se muestra un gráfico representativo de lo explicado anteriormente.

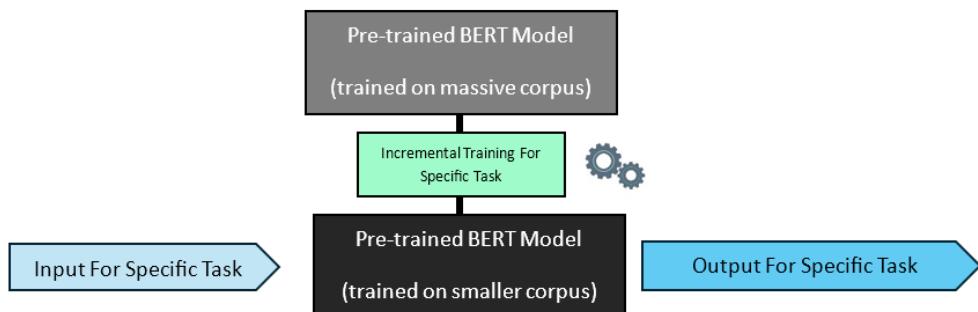


Figura 3 Transfer Learning. Propia.

2.6 Métricas de Evaluación

2.6.1 ICM - *Information Contrast Measure*

ICM (*Information Contrast Measure*) es una métrica de similitud basada en teoría de la información, diseñada para evaluar la precisión de las salidas de sistemas en problemas de clasificación, especialmente cuando se manejan estructuras jerárquicas y etiquetado múltiple. ICM se calcula por ítem, comparando los conjuntos de categorías asignados por el sistema con los del estándar de oro, teniendo en cuenta la especificidad estadística de las categorías (Amigó & Delgado, 2022).

El ICM-soft es una extensión del ICM que permite manejar tanto salidas de sistema suaves (*soft outputs*) como asignaciones suaves en el estándar de oro. En este contexto, se consideran las probabilidades asignadas a cada categoría en lugar de etiquetas binarias estrictas. La métrica evalúa la similitud entre estas distribuciones probabilísticas, adaptándose mejor a escenarios donde las categorías tienen asignaciones probabilísticas.

El ICM se define mediante el cálculo del contenido de información (IC) de las asignaciones de categorías. Formalmente, el IC de una categoría c se define en función de la probabilidad de encontrar un ítem etiquetado con c o más comúnmente. Para un documento d , el ICM compara los conjuntos de categorías asignados por el sistema $s(d)$ y el estándar de oro $g(d)$. La fórmula general de ICM es:

$$ICM(s(d), g(d)) = \sum_{c \in s(d) \cap g(d)} IC(c) - \sum_{c \in s(d) \cup g(d)} IC(c)$$

donde $IC(c)$ es el contenido de información de la categoría c .

El ICM-soft extiende el concepto de ICM al considerar asignaciones probabilísticas. El contenido de información para una categoría cc con un acuerdo v se define de manera inversa a la probabilidad de encontrar un ítem con un acuerdo igual o mayor a v para esa categoría. El cálculo de ICM-soft implica una función recursiva que estima el contenido de información de los conjuntos de asignaciones. La fórmula general de ICM-soft es:

$$ICM_soft(s(d), g(d)) = \sum_{c \in s(d) \cap g(d)} IC(c, v) - \sum_{c \in s(d) \cup g(d)} IC(c, v)$$

donde v representa el nivel de acuerdo.

2.6.2 Matriz de Confusión

La matriz de confusión es una Tabla que describe el rendimiento de un modelo de clasificación, mostrando el número de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

- **True Positives (TP):** El número de instancias que son positivas y que el modelo ha predicho correctamente como positivas.
- **True Negatives (TN):** El número de instancias que son negativas y que el modelo ha predicho correctamente como negativas.
- **False Positives (FP):** El número de instancias que son negativas pero que el modelo ha predicho incorrectamente como positivas.
- **False Negatives (FN):** El número de instancias que son positivas pero que el modelo ha predicho incorrectamente como negativas.

Véase ejemplificado en la Figura 4.

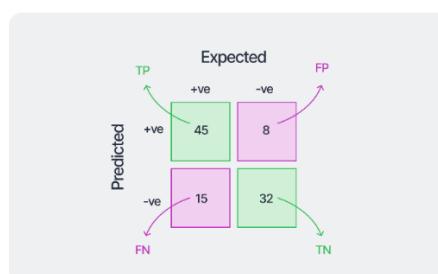


Figura 4 “Confusion Matrix for a binary class dataset” (v7labs, 2022)

En el caso de clasificación multiclas (caso de estudio en la Tarea 2 de EXIST 2024), la matriz de confusión tiene una fila y una columna para cada clase, mostrando cómo se clasificaron las instancias de cada clase en todas las clases posibles. Esto permite una evaluación más detallada del rendimiento del modelo en cada clase individualmente. Véase ejemplificado en la Figura 5.

		Expected			
		1	2	3	4
Predicted	1	52	3	7	2
	2	2	28	2	0
	3	5	2	25	12
	4	1	1	9	40

Figura 5 “Confusion Matrix for a multi-class dataset” (v7labs, 2022)

Aquí, los elementos diagonales (TP_1, TP_2, TP_3, TP_4) representan los verdaderos positivos para cada clase. Los elementos no diagonales (FP_1_2, FP_1_3, etc.) representan las predicciones incorrectas. Para cada clase individualmente:

- **True Positives (TP_x)**: El número de instancias de la clase x que han sido correctamente predichas como clase x.
- **False Positives (FP_x)**: El número de instancias que no son de la clase x pero que han sido incorrectamente predichas como clase x.
- **False Negatives (FN_x)**: El número de instancias de la clase x que han sido incorrectamente predichas como otra clase.
- **True Negatives (TN_x)**: El número de instancias que no son de la clase x y han sido correctamente predichas como no siendo de la clase x.

En resumen, estas métricas se derivan de la matriz de confusión, una herramienta que nos permite ver cómo se desempeña nuestro modelo de clasificación al comparar las predicciones del modelo con las etiquetas reales. La matriz de confusión nos ayuda a calcular TP, TN, FP y FN, que son esenciales para evaluar el rendimiento del modelo utilizando métricas como precisión, recall y F1-score.

2.6.3 Accuracy

La *accuracy* (precisión global) es la proporción de predicciones correctas sobre el total de predicciones realizadas.

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

2.6.4 Precision

La *precision* (precisión) es la proporción de verdaderos positivos sobre el total de predicciones positivas realizadas por el modelo.

$$\text{Precision} = \frac{TP}{TP+FP}$$

2.6.5 Recall

El *recall* (sensibilidad o tasa de verdaderos positivos) es la proporción de verdaderos positivos sobre el total de casos positivos reales.

$$\text{Recall} = \frac{TP}{TP+FN}$$

2.6.6 F1-score

El F1-score es la media armónica de la *precision* y el *recall*, proporcionando un equilibrio entre ambos.

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

2.6.7 Curva ROC (*Receiver Operating Characteristic*)

La curva ROC es una representación gráfica de la sensibilidad (*recall*) frente a la tasa de falsos positivos para diferentes umbrales de clasificación. En el eje x, se representa la tasa de falsos positivos (FPR), que es la proporción de instancias negativas incorrectamente clasificadas como positivas. En el eje y, se representa la tasa de verdaderos positivos (TPR), que es la proporción de instancias positivas correctamente clasificadas como positivas. La curva ROC traza TPR frente a FPR para diferentes umbrales de clasificación. Un modelo con un mejor rendimiento tendrá una curva ROC que se acerca más al vértice superior izquierdo del gráfico.

2.6.8 Area Under the Curve (ROC)

El área bajo la curva ROC (AUC) es un valor que resume la capacidad del modelo para distinguir entre clases. Un AUC de 1 indica un modelo perfecto, mientras que un AUC de 0.5 indica un modelo que no tiene capacidad de discriminación, equivalente a una clasificación aleatoria.

$$1 - \text{especificidad} = \frac{FP}{VN + FP}$$

2.7 Ensemble de Modelos

Un *ensemble* de modelos es una técnica en el aprendizaje automático que combina múltiples modelos individuales para mejorar la precisión en la predicción de una tarea específica. Estos modelos individuales pueden ser del mismo tipo o diferentes, y la idea es aprovechar las fortalezas de cada modelo y mitigar sus debilidades para obtener una predicción más precisa y robusta. Las estrategias comunes para construir un *ensemble* incluyen promediar predicciones, votar por la predicción más común y entrenar modelos adicionales sobre los residuos de modelos anteriores. La elección de la estrategia depende del problema y los modelos utilizados en el *ensemble*. En la Figura 6 se muestra gráficamente el funcionamiento de un *ensemble* de modelos.

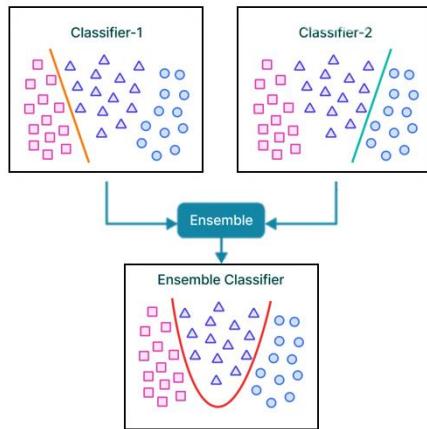


Figura 6 “Técnica de Ensemble de Modelos”

2.8 Aprendizaje con Desacuerdo (*Learning with Disagreement*)

El aprendizaje con desacuerdo es una estrategia en el aprendizaje automático que se basa en la idea de que los modelos que tienen diferentes opiniones sobre un conjunto de datos pueden ser más robustos y generalizables (Uma, y otros, 2021). En lugar de confiar en la predicción de un solo modelo, se aprovecha el desacuerdo entre múltiples modelos para mejorar la calidad de la predicción. Esta estrategia se puede implementar de diversas formas, como el entrenamiento de múltiples modelos con diferentes arquitecturas o inicializaciones y el uso de técnicas de *ensemble* para combinar las predicciones (Vitsakis, y otros, 2023). El objetivo es fomentar la diversidad entre los modelos para que se complementen entre sí y se reduzca el riesgo de sobreajuste. En el contexto de este proyecto, se aplica *Learning with Disagreement* para mejorar la precisión y robustez del sistema de clasificación de tweets sexistas. Se aplican técnicas de la siguiente manera:

- **Recolección de Datos con Desacuerdo:** Durante la anotación, se registra el desacuerdo entre anotadores en lugar de buscar un consenso simple, lo que permite crear un conjunto de datos que refleja la diversidad de opiniones.
- **Entrenamiento con Técnicas de *Learning with Disagreement*:** Se entrena un modelo de aprendizaje automático utilizando etiquetas suaves, que representan la distribución de opiniones de los anotadores. Esto permite al modelo manejar la incertidumbre y las diferentes interpretaciones de los datos.
- **Evaluación de la Incertidumbre del Modelo:** Durante la evaluación, el modelo proporciona una probabilidad en lugar de una clasificación binaria, permitiendo entender el nivel de certeza de cada predicción.

Implementar estas técnicas aprovecha la diversidad de opiniones para crear modelos más inclusivos y precisos, especialmente en tareas donde la subjetividad y el contexto son cruciales. Esto no solo mejora la calidad del modelo, sino que también refleja mejor la complejidad de la percepción humana en la identificación de contenido sensible como el sexismo en tweets.

2.9 Tecnologías y Recursos Utilizados

En esta sección se detallarán las tecnologías y recursos empleados para la ejecución y resolución del proyecto.



Python. Utilizado como el lenguaje de programación principal debido a su versatilidad y su amplio ecosistema de bibliotecas especializadas en aprendizaje automático y procesamiento de lenguaje natural.



Jupyter. Herramienta empleada para crear y compartir documentos que contienen código en tiempo real, ecuaciones, visualizaciones y texto narrativo. Facilitó la experimentación y visualización de resultados y se empleó para ejecutar los cuadernos que entrenaban los modelos en el equipo del laboratorio I2C.



HuggingFace Transformers. Biblioteca esencial para la implementación de modelos basados en *Transformers*, como BERT y RoBERTa, que se utilizaron para el procesamiento de lenguaje natural y la categorización de intenciones en tweets.



PyTorch. Framework de aprendizaje profundo utilizado para la creación y entrenamiento de modelos de redes neuronales. Su flexibilidad y eficiencia fueron cruciales para ajustar y optimizar los modelos.



Google Colab. Plataforma en la nube que proporciona acceso a unidades de procesamiento gráfico (GPU), específicamente la GPU NVIDIA T4. Esta GPU, con 16 GB de VRAM y soporte para Tensor Cores, aceleró significativamente las tareas de entrenamiento de modelos.



Optuna. Utilizado para la optimización de hiperparámetros, permitiendo encontrar las configuraciones más eficientes para los modelos implementados, mejorando su rendimiento.



GPU del laboratorio "I2C". Adicionalmente, empleé la GPU GeForce RTX 4070, que se encuentra instalada en un equipo del laboratorio del grupo de investigación “I2C” de la Universidad de Huelva, para tareas de entrenamiento que requerían recursos computacionales intensivos.



GitHub. Plataforma utilizada para el control de versiones y la organización de los cuadernos de Jupyter, código fuente y documentación del proyecto. Facilitó la colaboración y la gestión eficiente del desarrollo del proyecto.



LaTeX es un sistema de composición de textos que permite crear documentos de alta calidad tipográfica, especialmente utilizados en la producción de documentos científicos y académicos.

CAPÍTULO 3

Metodología, Experimentación y Resultados

3.1 Descripción de la Tarea

La tarea abordada en este proyecto forma parte de la competición internacional “EXIST 2024” (sEXism Identification in Social neTworks), una iniciativa de gran relevancia en el campo de la detección de comportamientos sexistas en redes sociales. EXIST 2024 está organizada por la “Conference and Labs of the Evaluation Forum” (CLEF, 2024).

La *Conference and Labs of the Evaluation Forum* (CLEF) es una iniciativa internacional dedicada a la evaluación y mejora de tecnologías de recuperación de información y procesamiento de lenguaje natural. Fundada en 2000, CLEF ha evolucionado para incluir una amplia gama de tareas y desafíos que abordan problemas actuales en el campo de la inteligencia artificial y la informática. CLEF se caracteriza por su enfoque en la creación de *benchmarks* rigurosos y su promoción de la colaboración y la innovación en la investigación de tecnologías de información.

CLEF tiene como misión principal fomentar el avance de la tecnología a través de la organización de evaluaciones comparativas (*benchmarking*) y *workshops* enfocados en el desarrollo de soluciones innovadoras para problemas complejos de recuperación de información y procesamiento de lenguaje natural. Sus objetivos incluyen:

- **Evaluación Continua:** Proveer una plataforma para la evaluación continua de tecnologías, permitiendo a los investigadores comparar y mejorar sus sistemas.
- **Innovación:** Fomentar la innovación mediante la introducción de nuevos desafíos y tareas que reflejan los problemas emergentes en el campo.
- **Colaboración:** Promover la colaboración internacional entre investigadores, facilitando el intercambio de ideas y técnicas.

CLEF se organiza en torno a varios componentes clave:

- **Conferencias Anuales.** Las conferencias de CLEF son eventos anuales que reúnen a investigadores de todo el mundo para discutir avances y desafíos en el campo de la recuperación de información y procesamiento de lenguaje natural. Estas conferencias incluyen presentaciones de trabajos de investigación, paneles de discusión y sesiones de pósteres.
- **Laboratorios (Labs).** Los *labs* de CLEF son grupos de trabajo dedicados a tareas específicas de evaluación. Cada lab define sus propias tareas y métricas de evaluación, proporcionando datos y directrices para los participantes. Estos *labs* cubren una variedad de temas, desde la recuperación de información multilingüe hasta la minería de opiniones y la detección de desinformación.

- **Workshops y Tutoriales.** Además de las conferencias y *labs*, CLEF organiza workshops y tutoriales que ofrecen formación y oportunidades de *networking* para investigadores y profesionales. Estos eventos son cruciales para la transferencia de conocimiento y la formación de nuevas colaboraciones.

A lo largo de los años, CLEF ha tenido un impacto significativo en el campo de la informática y la inteligencia artificial. Ha facilitado el desarrollo de nuevos algoritmos y técnicas que se han convertido en estándares en la industria. Además, CLEF ha ayudado a establecer una comunidad robusta de investigadores y profesionales que colaboran en la resolución de problemas complejos y en la mejora de la tecnología. En la Figura 7 se muestran las ediciones celebradas de CLEF hasta el momento.

 CLEF 2024 9-12 September 2024 Grenoble, France	 CLEF 2023 18-21 September 2023 Thessaloniki, Greece	 CLEF 2022 5-8 September 2022 Bologna, Italy	 CLEF 2021 21-24 September 2021 Bucharest, Romania (virtual)
 CLEF 2020 22-25 September 2020 Thessaloniki, Greece (virtual)	 CLEF 2019 9-12 September 2019 Lugano, Switzerland	 CLEF 2018 10-14 September 2018 Avignon, France	 CLEF 2017 11-14 September 2017 Dublin, Ireland
 CLEF 2016 5-8 September 2016 Evora, Portugal	 CLEF 2015 8-11 September 2015 Toulouse, France	 CLEF 2014 15-18 September 2014 Sheffield, UK	 CLEF 2013 23-26 September 2013 Valencia, Spain
 CLEF 2012 17-20 September 2012 Rome, Italy	 CLEF 2011 19-22 September 2011 Amsterdam, The Netherlands	 CLEF 2010 20-23 September 2010 Padua, Italy	 CLEF 2009 30 September-2 October 2009 Corfu, Greece
 CLEF 2008 17-19 September 2008 Aarhus, Denmark	 CLEF 2007 19-21 September 2007 Budapest, Hungary	 CLEF 2006 20-22 September 2006 Alicante, Spain	 CLEF 2005 21-23 September 2005 Vienna, Austria
 CLEF 2004 15-17 September 2004 Bath, UK	 CLEF 2003 21-22 August 2003 Trondheim, Norway	 CLEF 2002 19-20 September 2002 Rome, Italy	 CLEF 2001 3-4 September 2001 Darmstadt, Germany
 CLEF 2000 21-22 September 2000 Lisbon, Portugal			

Figura 7 “CLEF Editions” (CLEF, 2024)

En esta competición, me he centrado en la participación de dos tareas principales, denominadas *Task 1* y *Task 2*, que buscan identificar y clasificar mensajes sexistas en Twitter.

3.1.1 Task 1. Identificación de Sexismo

Esta tarea implica una clasificación binaria en la que los sistemas deben decidir si un tweet es sexista o no. Se proporcionan ejemplos de mensajes sexistas y no sexistas para ilustrar esta distinción:

- Ejemplo de mensaje sexista: "Mujer conduciendo, ¡ten cuidado!".
- Ejemplo de mensaje no sexista: "Acabo de ver a una mujer usando una mascarilla afuera, azotar a su perro atado muy firmemente y debo decir que me encanta aprender absolutamente todo sobre un extraño en un solo instante".

3.1.2 Task 2. Detección de la Intención del Autor

Esta tarea tiene como objetivo categorizar el mensaje según la intención del autor. Se propone una tarea de clasificación multiclase de entre estas tres clases posibles:

- **Direct (Mensaje Sexista directo).** La intención era escribir un mensaje que es sexista por sí mismo o que incita a ser sexista. Ejemplo:

"Una mujer necesita amor, para llenar la nevera, si un hombre puede darle esto a cambio de sus servicios (tareas domésticas, cocina, etc.), no veo qué más necesita ella".

- **Reported (Mensaje Sexista Reportado).** La intención es informar y compartir una situación sexista sufrida por una mujer o mujeres en primera o tercera persona. Ejemplo:

"Hoy, uno de los alumnos de mi clase de primer año no podía creer que había perdido una carrera contra una niña".

- **Judgemental (Mensaje de Juicio).** La intención es de juicio, ya que el tweet describe situaciones o comportamientos sexistas con el objetivo de condenarlos. Ejemplo:

"Siglo XXI y aún ganamos un 25% menos que los hombres #NoRenuncio".

3.2 Evaluación de Resultados

Para las tareas 1 y 2 de EXIST 2024, la evaluación se realiza en dos modos principales (Carrillo-de-Albornoz, y otros, 2024).

- **Hard-hard Evaluation:** En este modo, tanto la salida del sistema como el estándar de oro utilizan etiquetas binarias estrictas (*hard labels*). Se emplea ICM como métrica oficial, junto con el F1-score. Para derivar las etiquetas *hard* en el estándar de oro, se utiliza un umbral probabilístico basado en las anotaciones de los diferentes evaluadores. Por ejemplo, para la tarea 1, se selecciona la clase anotada por más de 3 anotadores.
- **Soft-soft Evaluation:** En este modo, se comparan las probabilidades asignadas por el sistema con las probabilidades del estándar de oro. Se utiliza ICM-soft como métrica oficial. Este enfoque permite una evaluación más fina de las salidas del sistema en contextos donde las categorías tienen asignaciones probabilísticas.

PyEvALL (The Python library to Evaluate ALL) es una herramienta de evaluación para sistemas de información que permite valorar una amplia gama de métricas, cubriendo diversos contextos de evaluación, incluyendo:

- Clasificación: Evaluación de modelos en términos de precisión, *recall*, F1-score, entre otros.
- Ranking: Medición de la calidad de sistemas de ordenación de elementos.
- LeWiDi (*Learning with Disagreement*): Evaluación en contextos con desacuerdo en las anotaciones.

PyEvALL se puede instalar mediante PIP¹ o desde el código fuente. Las instrucciones detalladas de instalación se encuentran en el archivo README del repositorio. La documentación de PyEvALL explica cómo utilizar la herramienta, incluyendo ejemplos de entrada y salida, y los formatos requeridos para los reportes de evaluación. PyEvALL soporta diversos formatos, facilitando la integración con diferentes herramientas y sistemas de anotación.

3.3 Descripción de los Datasets Originales

3.3.1 Composición

El conjunto proporcionado por la organización contiene 7958 tweets etiquetados, tanto en inglés como en español. El conjunto está dividido en un conjunto de entrenamiento (*training*), formado por 6920 tweets y un conjunto de desarrollo (*dev*) con 1038 tweets. Durante la fase de desarrollo se utilizó el conjunto *dev* para ir validando los resultados obtenidos por los modelos. La distribución entre ambos idiomas está equilibrada. Los conjuntos de datos se proporcionan en formato JSON.

Cada tweet está representado como un objeto JSON con varios atributos, incluyendo un identificador único para el tweet ("id_EXIST"), el idioma del texto ("lang"), el texto del tweet ("tweet"), el número de personas que han anotado el tweet ("number_annotators"), y atributos detallados sobre los anotadores, como sus identificadores únicos ("annotators"), género ("gender_annotators"), grupo de edad ("age_annotators"), etnia ("ethnicity_annotators"), nivel de estudio alcanzado ("study_level_annotators") y país de residencia ("country_annotators").

Adicionalmente, se incluyen conjuntos de etiquetas para cada una de las tareas específicas del dataset: "labels_task1" indica si el tweet contiene expresiones sexistas o se refiere a comportamientos sexistas y "labels_task2" registra la intención del autor del tweet.

Finalmente, el atributo "split" señala a qué subconjunto del dataset pertenece el tweet, ya sea de entrenamiento, desarrollo o prueba, diferenciando también por idioma. También se registra una etiqueta denominada "labels_task3", que pertenece a la *Task 3* de la competición, y en la que no he participado.

En la Figura 8 se muestra una instancia de los datos contenidos en el fichero "training.json".

¹ PIP (*Python Installs Packages*) es una herramienta de gestión de paquetes que permite instalar, actualizar y eliminar paquetes de software y sus dependencias desde el *Python Package Index (PyPI)* y otros índices de paquetes.

```
{
  "100001": {
    "id_EXIST": "100001",
    "lang": "es",
    "tweet": "@Thechiflis Ignora al otro, es un capullo.El problema con este youtuber denuncia el acoso... cuando no afecta a la gente de izquierdas. Por ejemplo, en su video sobre el gamergate presenta como \"normal\" el acoso que reciben Fisher, Anita o Zöey cuando hubo hasta amenazas de bomba.",
    "number_annotation": 6,
    "annotators": ["Annotator_1", "Annotator_2", "Annotator_3", "Annotator_4", "Annotator_5", "Annotator_6"],
    "gender_annotation": ["F", "F", "F", "M", "M", "M"],
    "age_annotation": ["18-22", "23-45", "46+", "46+", "23-45", "18-22"],
    "ethnicities_annotation": ["White or Caucasian", "Hispano or Latino", "White or Caucasian", "White or Caucasian", "White or Caucasian", "Hispano or Latino"],
    "study_levels_annotation": ["Bachelor's degree", "Bachelor's degree", "High school degree or equivalent", "Master's degree", "Master's degree", "High school degree or equivalent"],
    "countries_annotation": ["Italy", "Mexico", "United States", "Spain", "Spain", "Chile"],
    "labels_task1": ["YES", "YES", "NO", "YES", "YES", "YES"],
    "labels_task2": ["REPORTED", "JUDGEMENTAL", "-", "REPORTED", "JUDGEMENTAL", "REPORTED"],
    "labels_task3": [
      ["OBJECTIFICATION"],
      ["OBJECTIFICATION", "SEXUAL-VIOLENCE"],
      ["-"],
      ["STEREOTYPING-DOMINANCE"],
      ["SEXUAL-VIOLENCE"],
      ["IDEOLOGICAL-INEQUALITY", "MISOGYNY-NON-SEXUAL-VIOLENCE"]
    ],
    "split": "TRAIN_ES"
  }
},
```

Figura 8 Instancia de los datos contenidos en el fichero “training.json”

3.3.2 Proceso de Anotación

Cada anotador anota 60 tweets. Debe leer cada tweet cuidadosamente y responder a la pregunta principal sobre su contenido. “¿Es el tweet sexista en alguna forma, o describe situaciones en las que ocurre tal discriminación?” Si la respuesta a la pregunta principal es Yes, se responde a otra pregunta determinar la intención del autor del tweet. Intenciones posibles:

Direct: La intención es escribir un mensaje que es sexista por sí mismo. Ejemplo:

“A woman needs love, to fill the fridge, if a man can give this to her in return for her services (housework, cooking, etc), I don’t see what else she needs.”

Reported: La intención del autor es reportar o describir una situación o evento sexista sufrido por una mujer o mujeres en primera o tercera persona. Ejemplo: *“I doze in the subway, I open my eyes feeling something weird: the hand of the man sat next to me on my leg #SquealOnYourPig.”*

Judgemental: La intención del autor es ser crítico, describiendo situaciones o comportamientos sexistas con el objetivo de condenarlos. Ejemplo: *“As usual, the woman was the one quitting her job for the family’s welfare.”*

Cabe destacar que, en algunos casos de etiquetado, aparece la etiqueta UNKNOWN, la cual se asigna a aquellos tweets para los que el anotador en concreto no ha aportado ninguna etiqueta u opinión. Estas serán descartadas, ya que no son una clase que clasificar. Por ejemplo, en el dataset training.json hay 309 etiquetas UNKNOWN de las 124560 totales, representando el 0.24%.

3.3.3 Aplicación de Técnicas de *Learning with Disagreement*

El *dataset* está diseñado para trabajar con técnicas de aprendizaje con desacuerdo. Esto implica que múltiples anotadores han etiquetado cada tweet, proporcionando diversas perspectivas sobre el contenido. Los atributos detallados sobre los anotadores, como género, edad, etnia y nivel de estudios permiten analizar y ajustar el modelo considerando posibles sesgos en las anotaciones, facilitando un enfoque más inclusivo y representativo en el análisis del contenido.

3.4 Descripción General de la Metodología

En este proyecto, se ha seguido un enfoque riguroso y sistemático basado en los principios del determinismo y el método científico para garantizar la reproducibilidad y la validez de los resultados obtenidos; que viene descrito en la Figura 9. El determinismo en el contexto de experimentos computacionales se refiere a la capacidad de obtener los mismos resultados cada vez que se ejecuta un experimento bajo las mismas condiciones. Esto es crucial para la reproducibilidad, que permite que otros investigadores verifiquen los resultados y los comparén con otros trabajos. Para garantizar el determinismo en este proyecto, se han realizado varias acciones:

- **Definición de Semillas.** Se han establecido valores fijos de semillas en todas las librerías y entornos utilizados. Esto incluye la definición de semillas para la generación de números aleatorios en librerías como *random*, *numpy*, y *torch* (*PyTorch*). También se ha configurado *PyTorch* para asegurar el determinismo en sus operaciones.
- **Configuración de Entornos de Ejecución.** Se ha asegurado que la ejecución en GPU sea determinista configurando adecuadamente los parámetros y opciones en las librerías correspondientes.
- **Documentación y Configuración.** Se ha documentado y configurado el entorno de ejecución, incluyendo versiones de software, librerías y parámetros específicos utilizados durante el entrenamiento y evaluación de los modelos.



Figura 9 Método científico aplicado al proyecto. Propia

3.5 Selección de los Modelos *Transformers*

En esta sección se detallan los modelos de *Transformers* considerados para el proyecto, así como las razones de su selección.

- **XLM-RoBERTa Base:** Modelo multilingüe basado en RoBERTa, entrenado en 100 idiomas usando datos de CommonCrawl. Es ideal para tareas de comprensión del lenguaje natural en entornos multilingües, destacando por su capacidad para manejar diversidad lingüística (Conneau, y otros, 2020).
- **RoBERTa Base BNE:** Variante de RoBERTa optimizada con datos del Banco Nacional de España, centrada en el español. Este modelo se especializa en terminología y matices del español, particularmente en el dominio financiero, proporcionando una comprensión más precisa y específica para textos en este ámbito (Rodríguez, 2021).
- **DeBERTa v3 Base:** Mejora de BERT y RoBERTa, incorpora atención desentrelazada y codificación de posición mejorada. DeBERTa v3 se destaca por su rendimiento en múltiples benchmarks ofreciendo precisión avanzada en tareas de procesamiento del lenguaje natural (He, Liu, Gao, & Chen, 2020).
- **BERT Base Multilingual Uncased:** Modelo de BERT para múltiples idiomas, entrenado sin distinguir entre mayúsculas y minúsculas. Diseñado para aplicaciones multilingües, es robusto en el manejo de texto variado y diverso, siendo ampliamente utilizado en tareas globales de procesamiento del lenguaje natural (Devlin, Chang, Lee, & Toutanova, 2019).

Junto con el uso de estos *Transformers* se ha utilizado una técnica de *ensemble* para combinarlos tras aplicar las técnicas de *Learning with Disagreement*, de cara a obtener mejores rendimientos. Utilizar múltiples modelos con diferentes arquitecturas y datos de entrenamiento permite capturar una mayor variedad de patrones y características del lenguaje, mejorando la robustez y precisión del sistema.

3.6 Baseline

Un *baseline* es un punto de partida que se establece para comparar los resultados que se irán obteniendo durante el desarrollo con el objetivo de determinar su eficacia, y hacer notable el correcto progreso de los experimentos. La finalidad es establecer un nivel de rendimiento base que otros modelos deben superar para ser considerados mejores. Surgen, a raíz de este proceso, dos vías posibles para abordar ambas tareas (*Task 1* y *Task 2*), y dos posibles baselines (versión A y versión B). Será al final de este subcapítulo, donde se argumenta cuál ha sido la vía de desarrollo elegida. La Figura 10 plasma gráficamente lo anterior descrito. Ilustra dos enfoques de *baseline*:

1. Versión B Baseline:

- **Tarea 1: Clasificación Binaria:** El modelo clasifica un tweet como sexista ("Yes") o no sexista ("No").

- **Tarea 2: Clasificación Multiclasificación:** Si el tweet es clasificado como sexista, se clasifica en una de tres categorías específicas: "Direct", "Reported" o "Judgemental".

2. Versión A Baseline:

- **Clasificación Multiclasificación:** Un solo modelo clasifica directamente el tweet en una de las cuatro categorías posibles: "Direct", "Reported", "Judgemental" o "No".
- **Versión B:** Dos pasos de clasificación (primero binario, luego multiclasificación para sexistas).
- **Versión A:** Un solo paso de clasificación (multiclasificación para todas las etiquetas posibles).

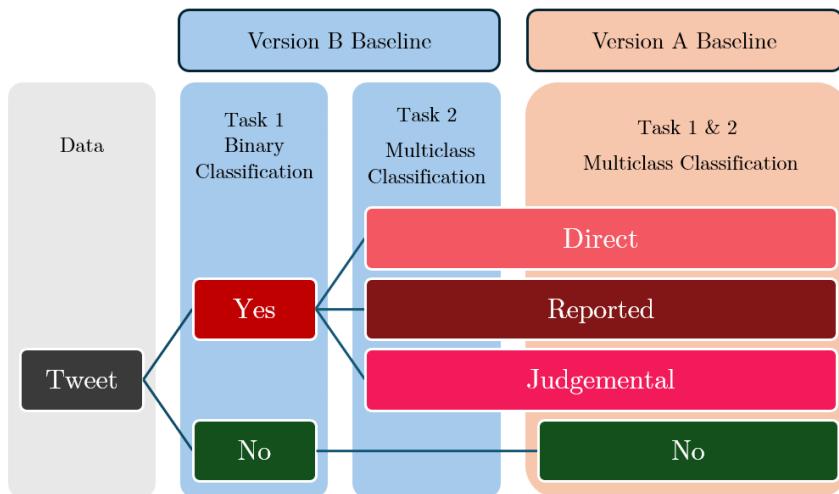


Figura 10 Clases posibles en la Task 1 y Task 2. Propia

3.6.1 Baseline de la versión A

En este punto nos centramos en la idea de entrenar modelos y clasificar información empleando un único clasificador multiclasificación, utilizando todas las etiquetas posibles a la vez: *No*, *Direct*, *Reported* y *Judgemental*. Por lo tanto, la primera idea es trabajar en un clasificador que clasifique las 4 clases a la vez. Para establecer el *baseline* de esta primera estrategia, se emplearon los conjuntos de datos que la competición entrega sin preprocessamiento alguno y los Transformers seleccionados descritos en la sección 3.4. Para el entrenamiento del *baseline* se utilizó un conjunto arbitrario de valores de hiperparámetros, concretamente los más utilizados en la literatura. Estos valores son: *batch size* de 32 (tamaño de lote para el entrenamiento), *learning rate* de 2e-5 (tasa de entrenamiento), *max length* de 128 (longitud máxima de las secuencias de entrada) y *weight decay* (decaimiento del peso para regularización) de 0.01. El número de épocas máximas de entrenamiento estaban limitadas a 10 con un *early stopping* establecido en tres épocas. Cabe destacar que se usaron todos los datos disponibles en ambos idiomas, español e inglés.

En la Tabla 1 se presentan los resultados para el *baseline* de la versión A tras el entrenamiento de los modelos pre-entrenados elegidos.

Tabla 1 Resultados para el baseline de la versión A

F1 macro	
Modelo entrenado	Resultado baseline A
XLM RoBERTa Base	0.4983
Deberta v3 Base	0.4910
Roberta Base Bne	0.4599
Bert Base Multilingual Uncased	0.4388

Como se puede observar, los resultados obtenidos mediante esta estrategia de clasificación son muy imprecisos y bajos. Por ejemplo, en el caso del modelo pre-entrenado XLM RoBERTa Base, el valor de F1 para la clase *No* (clase mayoritaria) fue de 0.8129, mientras que el valor de F1 para la clase *Judgemental* (clase minoritaria) fue de 0.3419. Esta distribución de resultados se repite para los demás modelos. Se puede concluir que los modelos se comportan correctamente con las clases mayoritarias, pero tienen un bajo rendimiento con las minoritarias.

3.6.2 Baseline de la versión B

Para la versión B, primero se clasificaron los Tweets en las clases de la *Task 1* (*Yes* y *No*), y después se clasificaron los Tweets catalogados como *Yes* anteriormente en las tres diferentes clases de la *Task 2* (*Direct*, *Reported* y *Judgemental*). A priori, esta es la mejor vía de ejecución del proyecto, dado que, en la evaluación de los resultados finales entregados a la competición, se penaliza mayor un error entre las clases *Yes* y *No*, que entre las diferentes clases de *Yes* (en lo que a la evaluación de la *Task 2* se refiere). En las Tablas 2 y 3, se muestran los resultados obtenidos en este “baseline” de la versión B”.

Tabla 2 Resultados para el baseline de la versión B en la primera clasificación, binaria

F1 macro	
Modelo entrenado	Resultado baseline B
XLM RoBERTa Base	0.7807
Deberta v3 Base	0.7820
Roberta Base Bne	0.7584
Bert Base Multilingual Uncased	0.7618

Tabla 3 Resultados para el baseline de la versión B en la primera clasificación, multiclas

F1 macro	
Modelo entrenado	Resultado baseline B
XLM RoBERTa Base	0.5683
Deberta v3 Base	0.5556
Roberta Base Bne	0.5305
Bert Base Multilingual Uncased	0.5292

Como se puede observar, los resultados mejoraron notablemente. Para sacar una comparativa con los resultados obtenidos en el *baseline* de la versión A, se promediaron los valores que se muestran en las Tablas 2 y 3 para obtener una medida F1 media. Estos resultados se muestran en la Tabla 4.

Tabla 4 Comparativa de resultados entre baselines A y B

Modelo entrenado	F1 Macro	Media Resultado baseline B	Resultado baseline A
XLM RoBERTa Base	0.6745	0.4983	
Deberta v3 Base	0.6688	0.4910	
Roberta Base Bne	0.6444	0.4599	
Bert Base Multilingual Uncased	0.6455	0.4388	

Analizando los valores para F1 obtenidos en ambas estrategias, se concluye que la estrategia “B” es la que mejor rendimiento de clasificación ha dado y por tanto fue la elegida para los experimentos posteriores.

Para la versión “B”, podemos observar cómo en la primera clasificación, los resultados son notablemente mejores que en la segunda. En la primera clasificación (en el caso de XLM RoBERTa Base) la F1 se sitúa en 0.7807, mientras que para la segunda clasificación en 0.5683.

Los modelos clasifican mejor la tarea binaria que la tarea multiclase. Si bien, más adelante, trataré este tema con profundidad, es importante dejar constancia de cómo se han comportado los modelos y desglosar un poco esta información en la segunda clasificación multiclase para la versión B del *baseline*.

La clase *Direct* se está clasificando mejor en comparación con las otras dos clases restantes. La razón es que esa clase es mayoritaria con respecto a las otras dos. En los próximos pasos, trataré de solucionar este problema, mediante el uso de técnicas de PLN.

En las Figuras 11, 12, 13 y 14 se presentan las cuatro matrices de confusión para la clasificación “multiclase” para la versión B del *baseline*, donde se aprecia el buen rendimiento de la clasificación para la clase *Direct* frente a un rendimiento más discreto de las clases *Reported* y *Judgemental*.

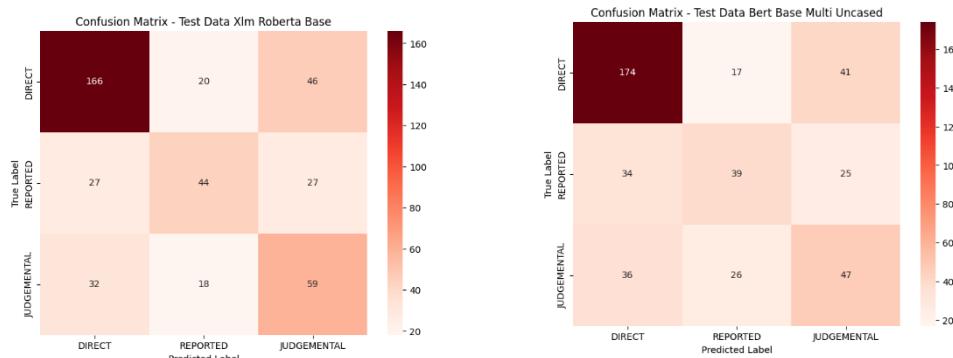


Figura 12 Matriz de confusión para Xlm Roberta Base, baseline B

Figura 11 Matriz de confusión para Bert Base M., baseline B

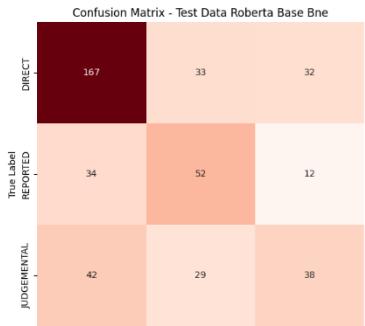


Figura 14 Matriz de confusión para Roberta Base Bne, baseline B

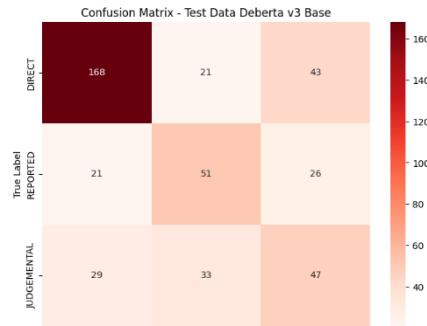


Figura 13 Matriz de confusión para Deberta v3 Base, baseline B

3.7 Preprocesamiento de Datos

El tratamiento de los datos y su correcto preprocesamiento es una parte fundamental de todo proyecto en procesamiento del lenguaje natural (PLN). El éxito de las aplicaciones y modelos en este campo depende en gran medida de la calidad y preparación de los datos utilizados. El preprocesamiento de datos en PLN abarca una serie de técnicas y metodologías que transforman los datos brutos en un formato adecuado para el análisis y modelado. Este proceso incluye la limpieza de datos, la tokenización, la eliminación de ruido, la normalización, el etiquetado de partes del discurso y la lematización, entre otros. En la Figura 15 se presenta el esquema que muestra la distribución y creación de los *datasets* que he empleado para entrenar a los modelos con los que he obtenido las predicciones para participar en la competición, tanto en la *Task 1* (anotado como v1.x) y en la *Task 2* (anotado como v2.x).

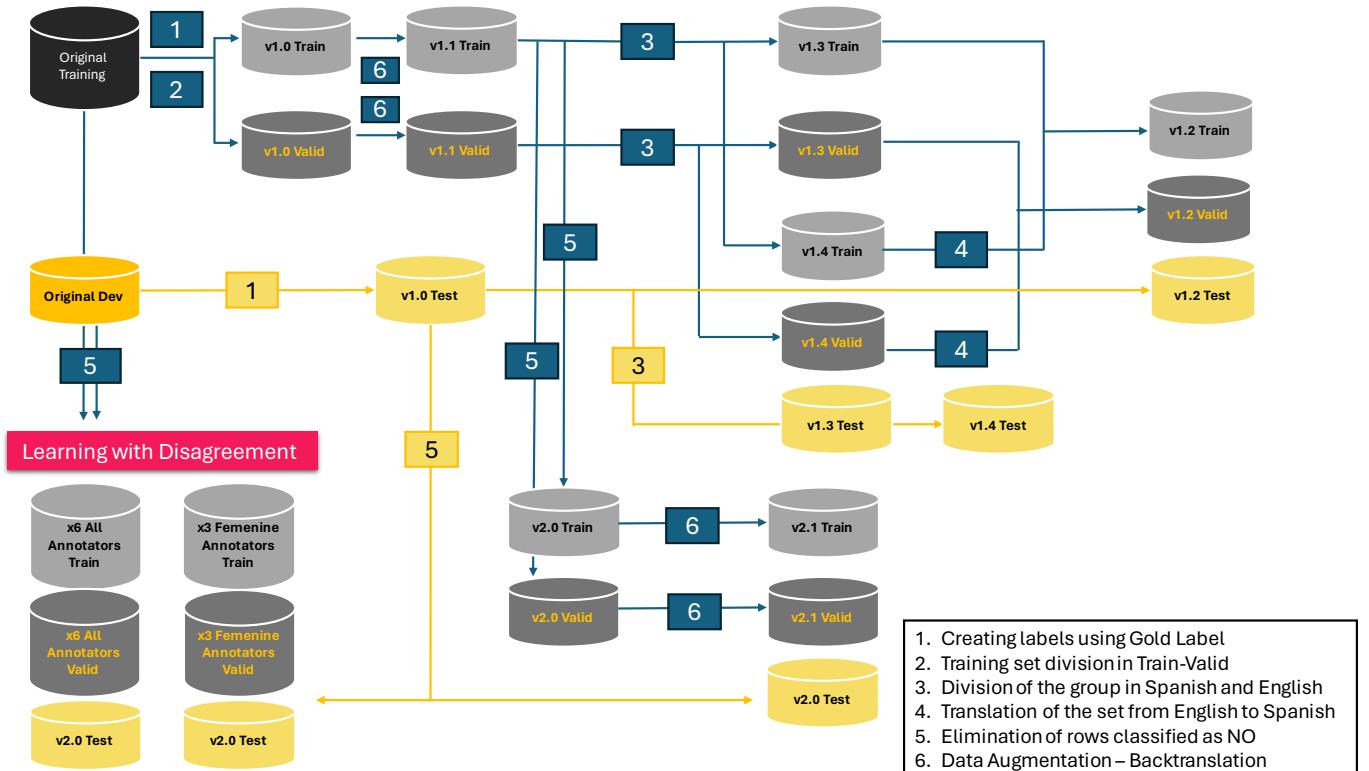


Figura 15 Datasets empleados para el entrenamiento de los modelos. Propia.

3.7.1 Descripción de los Datos

Los datasets originales que entrega la competición para el desarrollo de los modelos predictivos se distribuyen en dos ficheros; “*training.json*” y “*dev.json*”. En mi caso empleé el primero para crear los conjuntos de entrenamiento y validación para mis modelos, mientras que el segundo para poder evaluar el rendimiento de los modelos que iba entrenando. A su vez, también se entregó por parte de los organizadores el archivo “*test.json*” sobre el que más tarde se realizarán las predicciones por cada participante.

Anteriormente, en el capítulo “*3.2 Descripción de los Datasets Originales de Exist 2024*”, he descrito la naturaleza de la información, las etiquetas y clases y su proceso de anotación. Quiero centrarme, en este punto, en describir la distribución de clases de los *datasets* originales, describir ejemplos concretos de instancias, el tamaño medio de los mensajes para decidir el valor del *max_length* y describir los perfiles de los anotadores que han etiquetado las clases y su distribución.

Cada instancia es anotada por seis anotadores y, por tanto, contiene seis columnas por tarea con las opiniones de los anotadores y la información propia de cada anotador (género, edad, etnia, nivel de estudios y país). En la Tabla 5 se muestran instancias de ejemplo para la *Task 1* contenidas en “*training.json*”.

Tabla 5 Ejemplos de instancias para la Task 1

id_EXIST	lang	tweet	annotators IDs	labels_task1
101000	es	“No sean de esos que consiguen un peso y cambian con la gente. La plata no es culo.”	91, 92, 93, 94, 95, 96	“No”, “No”, “No”, “No”, “Yes”, “No”
201573	en	“@Avigeek96 Well men kill women everyday”	549, 550, 551, 552, 553, 554	“No”, “Yes”, “Yes”, “Yes”, “Yes”, “Yes”

Como se puede observar, las instancias vienen en dos idiomas, inglés y español. En esta Tabla 5 han sido omitidas varias columnas en las que se muestra la descripción de cada anotador (edad, género, nivel de estudios...) para que sea más legible. Se han incluido los identificadores de los anotadores que participan en la anotación de esas dos instancias. Cada anotador tiene su propio identificador y hay hasta 725 anotadores diferentes, pudiendo un mismo anotador etiquetar para varias instancias diferentes, pero no dos veces la misma.

La Tabla 6 muestra instancias de ejemplo para la *Task 2* contenidas en “*training.json*”. Como se puede observar, el Tweet 101000 ha sido etiquetado como no sexista por los anotadores 91, 92, 93, 94 y 96, mientras que el anotador 95 lo ha etiquetado como sexista (*Judgemental*).

Tabla 6 Ejemplos de instancias para la Task 2

id_EXIST	lang	tweet	annotators	labels_task2
101000	es	"No sean de esos que consiguen un peso y cambian con la gente. La plata no es culo."	91, 92, 93, 94, 95, 96	"-", "-", "-", "-", " "Judgemental", "-"
201573	en	"@Avigeek96 Well men kill women everyday"	549, 550, 551, 552, 553, 554	"-", "Reported", "Judgemental", "Judgemental", "Judgemental", "Reported"

Estas instancias han sido extraídas del fichero “training.json”, que cuenta con 6920 instancias, de las cuales 3660 son en español y 3260 son en inglés. En el caso del fichero “dev.json”, que cuenta con 1038 instancias totales, la distribución de idiomas es 549 para español y 489 para inglés.

También es importante estudiar la distribución de los diferentes perfiles de anotadores que etiquetan las instancias, para poder aplicar técnicas de *Learning with Disagreement*. Para el dataset “training.json” hay la misma cantidad de anotadores hombre que mujer, con un total de 20760 anotadores de cada género y un total de 41520. Para “dev.json” hay la misma distribución, 50% hombres (3114) y 50% mujeres (3114), de un total de 6228 anotadores. La distribución de edades también es equitativa en ambos datasets. Un tercio del total para cada grupo de edad (18-22 años, 23-45 años y mayor de 46 años). Para la distribución de etnias en el dataset “training.json” y “dev.json”, observamos una clara dominancia de la etnia “White or Caucasian” seguida de la clase “Hispano or Latino”. También, con respecto al nivel de estudios, la clase mayoritaria, es la de “Bachelor’s degree” y la minoritaria, “Less tan high school diploma” o “Doctorate”. Con respecto a la distribución de clases de etiquetas, la Figura 16 muestra la distribución.

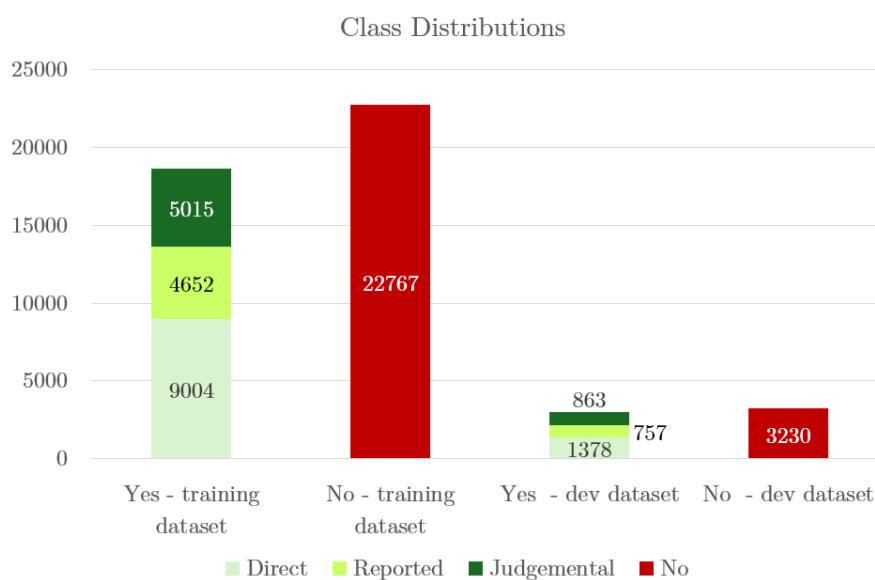


Figura 16 Distribución de clases de la Task 1 y Task 2 en los ficheros training y dev

Las distribuciones de clases en ambos conjuntos de datos, tanto para la tarea de clasificación binaria (*Yes* o *No*) como para la tarea de clasificación multiclas (*Direct*, *Reported*, *Judgemental*), muestran una consistencia significativa entre los conjuntos de entrenamiento y validación.

Este equilibrio es esencial para asegurar que los modelos entrenados no presenten un sesgo hacia una clase particular y puedan generalizar adecuadamente al conjunto de validación. Más adelante se implementarán técnicas de balanceo de clases como sobremuestreo de la clase minoritaria para el manejo del desequilibrio de clases.

3.7.2 Limpieza y Normalización

En el contexto del PLN, la limpieza y normalización de datos son pasos críticos para asegurar que los textos sean consistentes y estén libres de ruido antes de ser utilizados en modelos de aprendizaje automático. En concreto para la limpieza de los tweets se emplearon las siguientes técnicas y procesos de limpieza.

- **Conversión a Minúsculas.** Asegura que las palabras sean tratadas de manera uniforme, eliminando la distinción entre "Gato" y "gato". Simplifica el conjunto de datos y reduce el número de características únicas. Esto es particularmente importante para los modelos de PLN que distinguen entre palabras en función de su forma.
- **Eliminación de Enlaces.** Elimina cualquier enlace web presente en los tweets. Los enlaces no aportan valor semántico al contenido textual y a menudo no tienen relevancia directa para el análisis del sentimiento o el significado del texto.
- **Eliminación de Menciones a Usuarios.** Elimina menciones a otros usuarios y retweets del texto. Las menciones (@usuario) suelen no aportar información relevante al análisis semántico y pueden introducir ruido en los datos.
- **Eliminación de Hashtags.** Los hashtags pueden no ser relevantes para el análisis semántico y su eliminación ayuda a simplificar el texto, enfocando el análisis en las palabras y frases propiamente dichas.
- **Eliminación de Emojis.** Aunque los emojis pueden transmitir emociones o contextos, no son palabras y su interpretación puede ser compleja en el análisis textual. Inicialmente, consideré emplear una técnica de preprocesamiento que consistía en traducir los emojis a palabras literales. Sin embargo, esta técnica no mejoró los resultados, ya que, con frecuencia, la traducción literal carece de significado semántico en el contexto del texto. Por esta razón, se optó por eliminar los emojis para reducir el ruido y simplificar el análisis textual.

El antes y después de una instancia tras el proceso de limpieza y normalización descrito se puede ver ejemplificado en la Tabla 7. Las Tablas 8 y 9 muestran los resultados obtenidos tras esta fase de preprocesamiento de datos. Obsérvese la mejoría con respecto a los *baselines*.

Tabla 7 Limpieza y normalización de datos

Tweet original	Collab between WeAreEqual X @TaravaNFT ? 🤔 YOU ALREADY KNOW IT. 🙌 Join our Discord on how to join our exclusive Giveaway : 👉 #NFT #NFTGiveaway #art">https://t.co/x3stzfLLmh.#NFT #NFTGiveaway #art
Tweet limpio y normalizado	collab between weareequal x ? you already know it. join our discord on how to join our exclusive giveaway : .

Tabla 8 Resultados tras el preprocessamiento en los modelos de la Task 1

F1 Macro		
Modelo	Baseline	Preprocesamiento
XLM RoBERTa Base	0.7807	0.7893
Roberta Base Bne	0.7584	0.7595
Deberta v3 Base	0.7820	0.7865

Tabla 9 Resultados tras el preprocessamiento en los modelos de la Task 2

F1 Macro		
Modelo	Baseline	Preprocesamiento
XLM RoBERTa Base	0.5945	0.5990
Roberta Base Bne	0.4795	0.4817
Deberta v3 Base	0.5801	0.5854

3.7.3 Balanceo de Datos

En esta sección se detallan las técnicas de balanceo de datos con las que he experimentado durante el proyecto. El sobremuestreo es una técnica clave para abordar el desequilibrio de clases en conjuntos de datos. En PLN, el sobremuestreo con *backtranslation* implica traducir una oración a otro idioma y luego volver a traducirla al idioma original. Esto genera variaciones sintácticas y léxicas, aumentando la diversidad del conjunto de datos sin alterar su significado. La idea reside en aumentar las instancias de las clases *Reported* y *Judgemental*, para que los modelos aprendan mejor a clasificarlas. Para la tarea de traducción empleé Helsinki-NLP/opus, que es una colección de modelos de traducción automática creados por el Grupo de Procesamiento del Lenguaje Natural de la Universidad de Helsinki. Estos modelos forman parte del proyecto Open Parallel Corpus (OPUS) y están basados en la arquitectura de transformadores, utilizando específicamente la implementación MarianMT (Tiedemann & Thottingal, 2020). Para los tweets que están en idioma español, se traduce a de español a inglés, después de inglés a alemán y por último de alemán a español. En la Tabla 10 se presenta un ejemplo de una nueva instancia generada:

Tabla 10 Ejemplo de generación de datos mediante backtranslation en tweet en español

Tweet original	Se supone q me tengo q avergonzar d ser mamá? Jaja.jaja.jaja.naaaa
Nuevo tweet generado con Backtranslation	¿Debería avergonzarme de ser madre?

En los tweets que están en idioma inglés, se ha hecho una traducción de inglés a alemán, después de alemán a español y por último de español a inglés. En la Tabla 11 se presenta un ejemplo de una nueva instancia generada.

Tabla 11 Ejemplo de generación de datos mediante backtranslation en tweet en inglés

Tweet original	Easy to throw rocks and hide behind your gender or sexual identity #onhere
Nuevo tweet generado con backtranslation	Easy to throw stones and hide behind your sex or sexual identity #onhere

El sobremuestreo con *backtranslation* ha sido eficaz para mejorar la efectividad en las predicciones de los modelos.

3.7.4 Tokenización y Codificación

La tokenización y codificación son procesos esenciales en el preprocesamiento de textos para modelos de lenguaje basados en Transformers. Para ello se ha empleado la biblioteca *Transformers* de *Hugging Face*, concretamente la librería *Tokenizer* (*Hugging Face*, 2024). La tokenización consiste en dividir un texto en unidades menores, llamadas tokens. Los *Transformers* generalmente usan tokenizadores basados en subpalabras, como *Byte-Pair Encoding* (BPE) o *WordPiece*, que permiten manejar vocabularios de manera eficiente y gestionar palabras raras o desconocidas. Estos métodos dividen palabras en subpalabras comunes, lo cual reduce la cantidad de tokens desconocidos. Después de la tokenización, cada token se convierte en un índice numérico, conocido como *input id*, que el modelo puede procesar. Este proceso de conversión de tokens a índices se llama codificación. Los *input ids* corresponden a posiciones en un vocabulario predefinido que el modelo utiliza. En la Tabla número 12 se detalla este proceso mediante un ejemplo concreto.

Tabla 12 Proceso de tokenizado y codificado de una instancia

Al generar los *inputs ids*, los primeros números representan los tokens reales de la frase y los unos finales representan *padding* para ajustarse a una longitud fija de secuencia, un paso esencial para hacer las entradas compatibles con el modelo definido.

3.8 Ajuste de Hiperparámetros

El ajuste de hiperparámetros es una etapa crítica en el desarrollo de modelos de aprendizaje automático. Consiste en la selección de los valores óptimos para los parámetros que no se aprenden durante el entrenamiento, como la tasa de aprendizaje o el tamaño del lote de entrenamiento. Optuna permite definir un espacio de búsqueda de hiperparámetros y optimiza iterativamente para encontrar las mejores combinaciones que maximicen el rendimiento del modelo. La búsqueda exhaustiva, también conocida como *grid search*, es un método que explora sistemáticamente todas las combinaciones posibles de un conjunto predefinido de hiperparámetros. Aunque este método garantiza encontrar la combinación óptima dentro del espacio definido, su principal desventaja es el alto costo computacional, ya que el número de combinaciones crece exponencialmente con el número de hiperparámetros y los valores considerados para cada uno de ellos. Dado el alto costo computacional asociado con la búsqueda exhaustiva, he reducido el dataset de entrenamiento y validación en un 80% del original. Esto se hace para acelerar los experimentos y obtener resultados más rápidamente. Para implementar la búsqueda exhaustiva utilizando Optuna, se define un espacio de búsqueda de hiperparámetros, mostrado en la Tabla 13.

Tabla 13 Espacio de búsqueda de hiperparámetros definido

Hiperparámetro	Rango de Valores
Batch Size	[8, 16, 32]
Learning Rate	[3e-5, 5e-5]
Weight Decay	[0.001, 0.01, 0.1]
Max Length	[128]

El coste computacional de la búsqueda exhaustiva se deriva de la necesidad de entrenar y evaluar el modelo para cada combinación de hiperparámetros en el espacio definido. En este caso, el número total de combinaciones es 3 (batch sizes) * 2 (learning rates) * 3 (weight decays) = 18 combinaciones. En la Figura 17 se puede observar un ejemplo de las trazas de ejecución generadas para el modelo RoBERTa Base Bne, en la que se selecciona la mejor combinación.

Tras este proceso de optimización con todos los modelos empleados para la resolución de la *Task 1* y la *Task 2*, los resultados obtenidos fueron los siguientes. En la Tabla 14 se muestran los valores finales seleccionados de hiperparámetros para cada modelo.

```

Estudio de Hiperparámetros. Búsqueda: exhaustiva, Modelo: roberta

Iteración: 0, Valor: 0.8260047281323877, HP: {'per_device_train_batch_size': 16, 'learning_rate': 3e-05, 'weight_decay': 0.01}, Mejor: 0
Iteración: 1, Valor: 0.8035121025154248, HP: {'per_device_train_batch_size': 8, 'learning_rate': 3e-05, 'weight_decay': 0.01}, Mejor: 0
Iteración: 2, Valor: 0.81519557131041, HP: {'per_device_train_batch_size': 32, 'learning_rate': 5e-05, 'weight_decay': 0.001}, Mejor: 0
Iteración: 3, Valor: 0.836937256292095, HP: {'per_device_train_batch_size': 8, 'learning_rate': 5e-05, 'weight_decay': 0.1}, Mejor: 3
Iteración: 4, Valor: 0.8260047281323877, HP: {'per_device_train_batch_size': 32, 'learning_rate': 3e-05, 'weight_decay': 0.001}, Mejor: 3
Iteración: 5, Valor: 0.8367829686575599, HP: {'per_device_train_batch_size': 16, 'learning_rate': 5e-05, 'weight_decay': 0.1}, Mejor: 3
Iteración: 6, Valor: 0.8367829686575599, HP: {'per_device_train_batch_size': 32, 'learning_rate': 3e-05, 'weight_decay': 0.01}, Mejor: 3
Iteración: 7, Valor: 0.81519557131041, HP: {'per_device_train_batch_size': 16, 'learning_rate': 5e-05, 'weight_decay': 0.01}, Mejor: 3
Iteración: 8, Valor: 0.8367829686575599, HP: {'per_device_train_batch_size': 16, 'learning_rate': 3e-05, 'weight_decay': 0.1}, Mejor: 3
Iteración: 9, Valor: 0.8260047281323877, HP: {'per_device_train_batch_size': 32, 'learning_rate': 3e-05, 'weight_decay': 0.1}, Mejor: 3
Iteración: 10, Valor: 0.8035121025154248, HP: {'per_device_train_batch_size': 8, 'learning_rate': 3e-05, 'weight_decay': 0.1}, Mejor: 3
Iteración: 11, Valor: 0.8260869565217391, HP: {'per_device_train_batch_size': 8, 'learning_rate': 5e-05, 'weight_decay': 0.001}, Mejor: 3
Iteración: 12, Valor: 0.825344091124822, HP: {'per_device_train_batch_size': 8, 'learning_rate': 5e-05, 'weight_decay': 0.01}, Mejor: 3
Iteración: 13, Valor: 0.8573303113443875, HP: {'per_device_train_batch_size': 16, 'learning_rate': 5e-05, 'weight_decay': 0.01}, Mejor: 13
Iteración: 14, Valor: 0.8367829686575599, HP: {'per_device_train_batch_size': 16, 'learning_rate': 5e-05, 'weight_decay': 0.001}, Mejor: 13
Iteración: 15, Valor: 0.8240076518412243, HP: {'per_device_train_batch_size': 32, 'learning_rate': 5e-05, 'weight_decay': 0.1}, Mejor: 13
Iteración: 16, Valor: 0.8035121025154248, HP: {'per_device_train_batch_size': 8, 'learning_rate': 3e-05, 'weight_decay': 0.001}, Mejor: 13
Iteración: 17, Valor: 0.8257575757575758, HP: {'per_device_train_batch_size': 16, 'learning_rate': 3e-05, 'weight_decay': 0.001}, Mejor: 13

```

Figura 17 Estudio de Hiperparámetros. Búsqueda: exhaustiva, Modelo: Roberta

Tabla 14 Parámetros seleccionados tras la optimización

Hiperparámetro	XLM RoBERTa	RoBERTa B. Bne	DeBERTa v3 B.	BERT Base M.U.
Batch Size	16	8	16	8
Learning Rate	5e-5	5e-5	5e-5	3e-5
Weight Decay	0.01	0.01	0.1	0.01
Max Length	128	128	128	128

En referencia a las métricas obtenidas tras la optimización de hiperparámetros y la aplicación de las técnicas anteriormente explicadas, las Tablas 15 (para la *Task 1*) y 16 (para la *Task 2*) muestran los resultados obtenidos tras la optimización de hiperparámetros. Se puede observar una mejora considerable de los resultados utilizando los mejores valores de los hiperparámetros.

Tabla 15 Resultados para los modelos de la Task 1 tras ajustar hiperparámetros

F1 macro		
Modelo	Baseline	Tras optimizar hiperparámetros
XLM RoBERTa Base	0.7807	0.7876
Deberta v3 Base	0.7820	0.7871
Roberta Base Bne	0.7584	0.7616

Tabla 16 Resultados para los modelos de la Task 2 tras ajustar hiperparámetros

F1 macro		
Modelo	Baseline	Tras optimizar hiperparámetros
XLM RoBERTa Base	0.5945	0.6095
Deberta v3 Base	0.5801	0.5968
Roberta Base Bne	0.4795	0.4905

3.9 Entrenamiento y Selección de Modelos Finales

En este proyecto, se utilizaron dos tipos de tareas de clasificación para tweets: la primera tarea (*Task 1*) es una clasificación binaria para identificar tweets sexistas, y la segunda tarea (*Task 2*) es una clasificación multiclas para determinar la intención detrás de los tweets sexistas.

3.9.1 Configuración General de los Entrenamientos

Para llevar a cabo estas tareas, se utilizaron los pre-entrenados XLM-RoBERTa-Base, RoBERTa - Base-Bne y DeBERTa v3 Base de Hugging Face, implementado mediante la clase *AutoModelForSequenceClassification*. La configuración específica del entrenamiento se realizó utilizando la clase *Trainer* de Hugging Face, la cual facilita la configuración y ejecución del entrenamiento del modelo con hiperparámetros optimizados.

Se definieron varios hiperparámetros para el entrenamiento de cada modelo. Estos valores fueron establecidos según el proceso de “*Ajuste de Hiperparámetros*” ejecutado anteriormente para cada modelo. Se utilizó el optimizador *adamw_torch*, una versión optimizada del algoritmo Adam adaptada para PyTorch, para actualizar los pesos del modelo durante el entrenamiento.

La estrategia de evaluación se configuró para realizarse al final de cada época, lo que permite evaluar el rendimiento del modelo en el conjunto de validación de manera periódica y consistente. Asimismo, se configuró una estrategia de guardado para almacenar el modelo al final de cada época, con un límite total de guardado de 3 modelos para optimizar el uso del espacio de almacenamiento. Se habilitó la opción de cargar el mejor modelo al final del entrenamiento, basado en la métrica F1, para asegurar que el modelo final sea el que mejor rendimiento tiene en el conjunto de validación. Por ejemplo, en la Figura 18, se puede observar cómo es el proceso de entrenamiento para el modelo XLM-RoBERTa-Base de la versión v1.1.

```
Some weights of XLMRobertaForSequenceClassification were not initialized from the model checkpoint at FacebookAI/xlm-roberta-base and are newly initialized: ['classifier.dense.bias', 'classifier.dense.weight', 'classifier.out_proj.bias', 'classifier.out_proj.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
[810/1620 09:37 < 09:38, 1.40 it/s, Epoch 5/10]
```

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall	Auc	F1 Minoritaria	F1 Mayoritaria	Prec Rec
1	No log	0.435237	0.800000	0.799985	0.807694	0.808801	0.808801	0.798226	0.801743	0.693314
2	No log	0.426154	0.842857	0.841782	0.840867	0.843992	0.843992	0.828743	0.854822	0.752266
3	No log	0.456287	0.832967	0.828173	0.837617	0.824569	0.824569	0.799472	0.856874	0.754266
4	0.421000	0.511335	0.835165	0.834705	0.835413	0.839506	0.839506	0.825986	0.843424	0.738591
5	0.421000	0.646230	0.823077	0.817506	0.828783	0.813702	0.813702	0.785619	0.849392	0.741910

```
: TrainOutput(global_step=810, training_loss=0.33728135662314335, metrics={'train_runtime': 579.3668, 'train_samples_per_sec': 88.959, 'train_steps_per_second': 2.796, 'total_flos': 1205061994657080.0, 'train_loss': 0.33728135662314335, 'epoch': 5.0})
```

Figura 18 Train output para XLM RoBERTa Base de la versión v1.1

El proceso de entrenamiento fue monitoreado utilizando técnicas como el *EarlyStoppingCallback*, que detiene el entrenamiento si no se observa mejora en la métrica de validación después de tres

épocas consecutivas. En la imagen anterior puede observarse que tras la F1 obtenida en la segunda época de entrenamiento (0.8417), esta no se mejora en las tres épocas siguientes, por lo que el entrenamiento se detiene.

3.9.2 Identificación de Sexismo en Tweets – Task 1

Para el entrenamiento de los modelos finales que generarán las predicciones sobre los datos de test que entregó la competición para la *Task 1*, se seleccionaron los dos mejores en cuanto a mejores métricas obtenidas (F1 – score, ICM e ICM-soft) durante el proceso de entrenamiento.

Para entrenar el modelo de la versión v1.1, utilicé los datos del conjunto v1.1. Cada tweet en este conjunto está etiquetado por seis anotadores tanto en el conjunto de entrenamiento como en el de validación. Para obtener la etiqueta mayoritaria, siguiendo las directrices de la competición para obtener la llamada *gold label*, promedié las votaciones, seleccionando las etiquetas que recibieron dos o más votos de entre los seis anotadores posibles. En caso de empate, excluí completamente la instancia en cuestión. Se entrenó un modelo multilingüe, XLM-RoBERTa-Base, para manejar tanto las instancias en español como en inglés simultáneamente. En la Figura 19 se muestra una descripción de este proceso.



Figura 19 Flujo de entrenamiento del modelo v1.1

Se utilizó este modelo para predecir las etiquetas de los datos en el conjunto de prueba oficial de la competición. Los resultados se presentan indicando la etiqueta predicha para cada instancia del conjunto de prueba, seguida del *score_label*, que representa la puntuación de similitud que el clasificador asigna a la etiqueta predicha en una escala de 0 a 1.

Para obtener la *hard label*, seleccioné la etiqueta predicha. En cuanto a la *soft label*, dado que se trata de un clasificador de dos clases (*Yes* o *No*), asigné el valor de *score_label* a la clase mayoritaria en cada caso y calculé el valor de la clase minoritaria como 1 menos el *score_label*. Es importante destacar que la suma de los valores de las etiquetas en los resultados soft no debe exceder 1.

El proceso de entrenamiento del modelo v1.2 es prácticamente igual al proceso anterior descrito, pero con algunas diferencias clave en cuanto a los modelos utilizados y la estructura del flujo de trabajo. Para el entrenamiento, se utilizaron dos conjuntos de datos: v1.3 para las instancias en inglés y v1.4 para las instancias en español. En cuanto a los modelos, se emplearon DeBERTa v3 Base para el inglés y RoBERTa Base Bne para el español. En la Figura 20 se puede observar este proceso.

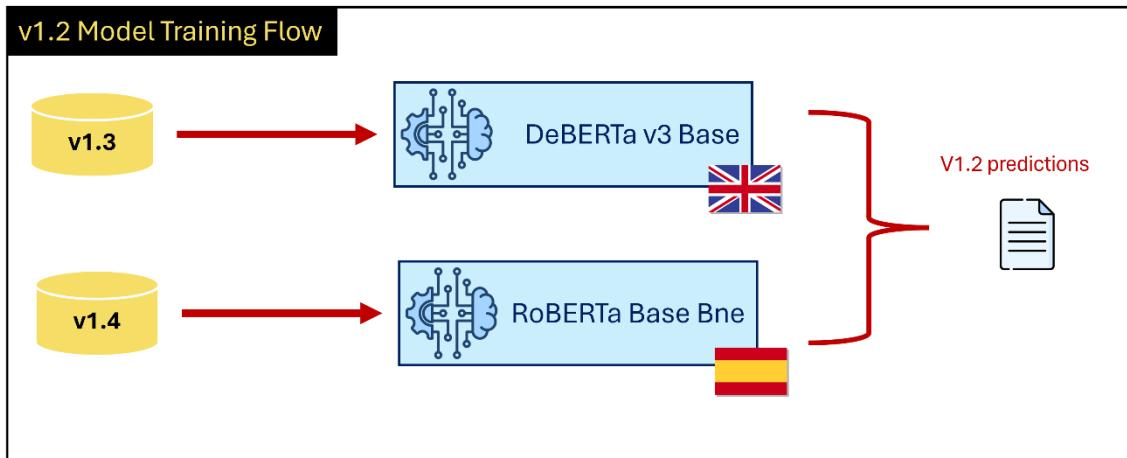


Figura 20 Flujo de entrenamiento del modelo v1.2

El flujo de trabajo comenzó con el entrenamiento separado de los dos modelos: el modelo DeBERTa v3 Base se utilizó para las instancias en inglés del conjunto de datos v1.3, y el modelo RoBERTa Base Bne se empleó para las instancias en español del conjunto de datos v1.4. Después de entrenar los modelos, se procedió a la predicción y generación de resultados.

3.9.3 Clasificación de la Intención en Tweets Sexistas – Task 2

El modelo v2.1 se diseñó para abordar la segunda tarea de la competición, la cual se enfoca clasificar la intencionalidad de los tweets categorizados anteriormente como sexistas por el modelo v1.2 (*Source Intention in Tweets*). Esta tarea sigue a la clasificación inicial de mensajes sexistas y busca categorizar dichos mensajes según la intención del autor, proporcionando así información sobre el rol que juegan las redes sociales en la emisión y difusión de mensajes sexistas. En esta tarea, se propone una clasificación multiclas entre tres clases *Direct*, *Reported* y *Judgemental*.

Los datos de entrenamiento provienen del dataset v2.1 y que contiene únicamente instancias de las 3 clases, eliminando las instancias categorizadas como *No*; evitando así introducir ruido en los datos de entrenamiento y afinando la precisión del modelo. Aquí sólo se generaron los datos *hard label* para las predicciones finales, debido a que el modelo no devuelve el *score label* de las clases predichas como minoritarias. Al ser tres clases posibles en vez de dos, no se puede hacer el complemento a uno como se hizo anteriormente, para obtener los valores *soft* de las otras dos clases. En la Figura 21 se puede observar este proceso.

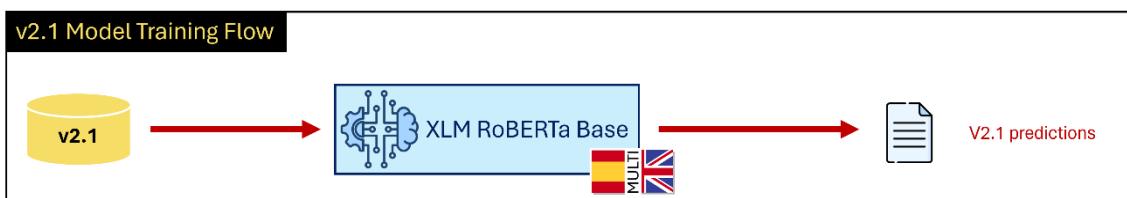


Figura 21 Flujo de entrenamiento del modelo v2.1

El siguiente modelo aplica *Learning with Disagreement* porque considera y aprovecha las diferencias de opinión entre múltiples anotadores humanos al etiquetar los datos de entrenamiento. Este enfoque permite capturar una mayor diversidad de perspectivas, lo que es especialmente útil en tareas subjetivas o complejas donde puede haber un desacuerdo significativo sobre las etiquetas correctas.

Este método mejorará las predicciones del modelo porque al integrar múltiples puntos de vista, se crea un conjunto de datos de entrenamiento más robusto y representativo. Además, las etiquetas suaves (*soft labels*) que resultan de este proceso permiten al modelo capturar la incertidumbre y variabilidad inherente en las anotaciones humanas, llevando a una mejor generalización y rendimiento en situaciones del mundo real donde los datos pueden no ser claros o estar completamente definidos. En la Figura 22 se puede observar este proceso.

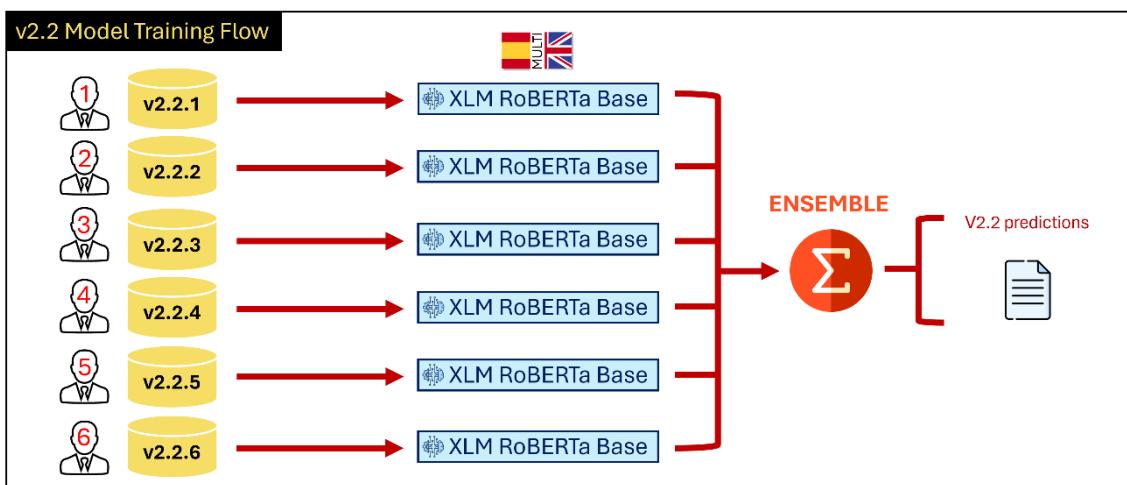


Figura 22 Flujo de entrenamiento del modelo v2.2

El flujo de entrenamiento del modelo mostrado en la imagen puede explicarse en detalle, centrándose en cómo se maneja el desacuerdo entre anotadores y cómo se generan las etiquetas suaves (*soft labels*). A continuación, se proporciona la explicación detallada de manera secuencial.

Los datos de entrenamiento provienen de seis grupos de anotadores diferenciados por género y edad: ["F 18-22", "F 23-45", "F 46+", "M 46+", "M 23-45", "M 18-22"]. Cada grupo de anotadores ha proporcionado etiquetas para los datos de entrenamiento. Seis conjuntos de datos (v2.2.1, v2.2.2, v2.2.3, v2.2.4, v2.2.5 y v2.2.6) se utilizan para entrenar seis instancias del modelo XLM RoBERTa Base.

Cada conjunto de datos corresponde a las anotaciones de uno de los seis grupos mencionados. Los seis modelos entrenados se combinan usando un método de ensemble. Este proceso integra las salidas de los diferentes modelos para producir una predicción final más robusta. El ensemble calcula un promedio ponderado (suma) de las predicciones de los seis modelos. Para generar las etiquetas suaves, se toma en cuenta la proporción de anotadores que votaron por cada etiqueta. Por ejemplo, si 2 de 6 anotadores etiquetaron un dato como *Direct*, la *soft label* para *Direct* sería $2/6 = 0.3333$. Este proceso se repite para las otras etiquetas, *Reported* y *Judgemental*.

En la tarea previa (*Task 1*), los datos fueron clasificados en las clases *Yes* y *No*. Si un dato fue clasificado como *Yes* con una probabilidad de 0.80, este valor se utiliza para ajustar las etiquetas suaves de *Task 2*. Por ejemplo, si la etiqueta suave para *Direct* es 0.33333, el valor ajustado sería $0.33333 * 0.80 = 0.26666$. Este ajuste se realiza para todas las subclases de *Yes* (*Direct*, *Reported* y *Judgemental*).

Este proceso hay que hacerlo para la etiqueta *Yes* cuando es clase mayoritaria en la *Task 1*, como para predecir el porcentaje de esta cuando es clase minoritaria en la *Task 1*. Cómo conclusión, se están calculando igualmente la ínfima probabilidad que las diferentes clases de *Yes* en las instancias que han sido clasificadas por los modelos de la versión 1 como *No*.

Por último, el clasificador de la versión v2.3 sigue las mismas directrices que el de la versión v2.2, explicado anteriormente, pero los datos de entrenamiento provienen de tres grupos de anotadores diferenciados por género y edad: ["F 18-22", "F 23-45", "F 46"]. Se han seleccionado los grupos únicamente femeninos para realizar los entrenamientos de los modelos que compondrán el *ensemble*. La Figura número 23 muestra una descripción del proceso descrito anteriormente para versión v2.3.

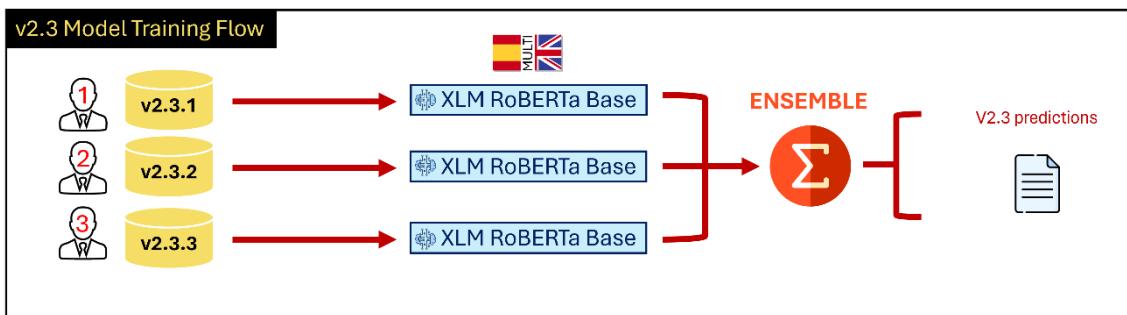


Figura 23 Flujo de entrenamiento del modelo v2.3

3.10 Validación y Evaluación de los Modelos

En esta sección se presentan los resultados de la validación y evaluación de los modelos entrenados en diferentes versiones durante la fase de desarrollo. La versión v1.x pertenece a los modelos finales entrenados para resolver la *Task 1* y los v2.x los de la *Task 2*.

3.10.1 Modelos de la versión v1.1

La versión v1.1 incluye clasificadores binarios para las etiquetas *Yes* y *No* de la *Task 1*, abarcando tanto inglés como español simultáneamente. En la Tabla 17 se pueden ver reflejados sus rendimientos.

Tabla 17 Validación y evaluación de los modelos de la versión v1.1

Versión v1.1

Modelo	ICM	ICM - NORM	F1
XLM RoBERTa Base	0.5501	0.7751	0.8500

3.10.2 Modelos de las versiones v1.3, v1.4 y v1.2

La versión v1.3 evaluó los modelos entrenados exclusivamente para inglés, destacándose DeBERTa v3 Base con un F1 macro de 0.8589. En la versión v1.4, se centraron en predicciones en español, logrando RoBERTa Base Bne el más alto F1 macro de 0.8630. La versión v1.2 combinó modelos optimizados para cada idioma, usando DeBERTa v3 Base para inglés y RoBERTa Base Bne para español, obteniendo una Media Final de 0.8617 en F1 macro, subrayando la importancia de la adaptación lingüística para una clasificación precisa de tweets multilingües. Cada modelo entrenado generó predicciones individuales para los tweets en su respectivo idioma. En las Tablas 18, 19 y 20 se pueden ver reflejados sus rendimientos.

Tabla 18 Validación y evaluación de los modelos de la versión v1.3

Versión v1.3 – predicciones inglés			
Modelo	ICM	ICM - NORM	F1 macro
XLM RoBERTa Base	-0.2581	0.3708	0.8540
DeBERTa v3 Base	-0.2515	0.3741	0.8589

Tabla 19 Validación y evaluación de los modelos de la versión v1.4

Versión v1.4 – predicciones español			
Modelo	ICM	ICM - NORM	F1 macro
XLM RoBERTa Base	-0.2208	0.3895	0.8262
RoBERTa Base Bne	-0.1630	0.4184	0.8630
BERT Base Uncased	-0.2347	0.3825	0.818

Tabla 20 Validación y evaluación de los modelos de la versión 1.2

Versión v1.2 – predicciones inglés + español			
Modelo	ICM	ICM - NORM	F1 macro
DeBERTa v3 base	0.5686	0.7753	0,8589
RoBERTa Base Bne	0.6012	0.8099	0,8630
Media Final	0.5849	0.7926	0.8617

3.10.3 Modelo de la versión v2.1

En la versión v2.1, el modelo XLM RoBERTa Base se utiliza como un clasificador multiclas para las tres categorías dentro de la etiqueta Yes. Este modelo es entrenado utilizando la gold_label oficial proporcionada por la organización. Los resultados de su evaluación son los siguientes:

- **ICM:** 0.1221
- **ICM - NORM:** 0.5238
- **F1 macro:** 0.5008

3.10.4 Modelos de la versión v2.2

En la versión v2.2, se evaluaron modelos utilizando seis grupos de anotadores distintos, junto con un ensemble. El rendimiento de estos tras la evaluación se presenta en la Tabla 21.

Tabla 21 Validación y evaluación de los modelos de la versión 2.2

Modelo	ICM	ICM - NORM	F1 macro
XLM RoBERTa [Ann_1]	-	-	0.5755
XLM RoBERTa [Ann_2]	-	-	0.5460
XLM RoBERTa [Ann_3]	-	-	0.5086
XLM RoBERTa [Ann_4]	-	-	0.5076
XLM RoBERTa [Ann_5]	-	-	0.5167
XLM RoBERTa [Ann_6]	-	-	0.5088
Ensemble	0.221	0.5476	0.5272

3.10.5 Modelos de la versión v2.3

En la versión v2.3, se evaluaron modelos utilizando únicamente tres grupos de anotadores femeninos, además de un ensemble. El rendimiento de estos tras la evaluación se presenta en la Tabla 22.

Tabla 22 Validación y evaluación de los modelos de la versión 2.3

Modelo	ICM	ICM - NORM	F1 macro
XLM RoBERTa [Ann_1]	-	-	0.5755
XLM RoBERTa [Ann_2]	-	-	0.5460
XLM RoBERTa [Ann_3]	-	-	0.5086
Ensemble	0.1709	0.511	0.5434

3.11 Análisis de Errores

En esta sección se expone un análisis detallado de algunos de los errores cometidos por los modelos en las predicciones para *Task 1* y la *Task 2*. Al examinar las clasificaciones erróneas, se pretende identificar patrones e *insights* sobre los desafíos que enfrentan los modelos. Por ejemplo, en la siguiente Tabla 23 se presentan algunos errores cometidos en la primera clasificación binaria.

Tabla 23 Análisis de errores en instancias de la Task 1

Tweet	Labels	Predictions
Woman driving beside me a few minutes ago holding her phone to her ear with her shoulder, while holding a mug of coffee. Baby on Boardsticker on both rear windows.	No	Yes
Por qué todos los hombres cuando su novia o esposa está embarazada andan más de culeros que de costumbre	Yes	No

En la primera instancia, el modelo puede haber interpretado la mención de una mujer realizando una acción imprudente como una crítica sexista hacia las mujeres en general. No comprendió que el comentario era una observación sobre una conducta específica y no una generalización sobre las mujeres. Para la segunda instancia, modelo no logró captar la generalización negativa sobre el comportamiento de los hombres, una característica común en comentarios sexistas. El uso de un lenguaje coloquial y vulgar puede haber confundido al modelo, haciéndole pasar por alto el contenido discriminatorio. A continuación, se exponen dos exemplificaciones para las instancias de la *Task 2* en la siguiente Tabla 24.

Tabla 24 Análisis de errores en instancias de la Task 2

Tweet	Labels	Predictions
Lo irónico es que en su mayoría sean hombres quienes apoyan la criminalización de las mujeres frente al aborto. Claro, a las mujeres hay que castigarlas, juzgarlas y señalarlas siempre, como si no fuera suficiente tener que cargar con el peso de una violación.	<i>Reported</i>	<i>Direct</i>
En total delirio esta tipa quiere legalizar el terrorismo. ¿Y esta escoria quiere definir los destinos de Chile? Permitirlo es de anti chilenos.	<i>Direct</i>	<i>No</i>

Para la primera instancia se presenta un ejemplo en el que el tweet tiene un tono fuerte y emotivo, destacando la injusticia hacia las mujeres. El modelo pudo haber interpretado esto como un ataque directo en lugar de una denuncia de discriminación. En la segunda instancia, el modelo no detectó el lenguaje despectivo y ofensivo (“tipa”, “escoria”) y el tono agresivo dirigido a una persona específica. La discusión política puede haber desviado la atención del modelo, llevándolo a no reconocer el ataque personal en el tweet.

3.12 Resultados Oficiales de la Competición

En esta sección se presentan los resultados oficiales obtenidos tras la participación en EXIST 2024. Las evaluaciones se llevaron a cabo utilizando tanto salidas y verdades de referencia duras (*hard-hard*) como suaves (*soft-soft*), y se aplicaron métricas específicas para cada tipo. Cada equipo tiene permitido presentar un total de tres “runs” por tipo de evaluación (*hard-hard* o *soft-soft*) y por tarea.

3.12.1 Ejemplos de Instancias en las *Runs* Presentadas

En esta sección se incluyen ejemplos de las instancias utilizadas en las “runs” presentadas. A continuación, se muestran dos capturas: una del archivo JSON proporcionado por la competición con las instancias a predecir (Figura 24) y otra de uno de los archivos que se envía a la competición con esas instancias ya predichas (Figura 25).

```
{
    "500001": {
        "id_EXIST": "500001",
        "lang": "es",
        "tweet": "@Eurogamer_es Todo gamergate desde el desarrollo hasta los foros de juegos, clásico del mundo de los videojuegos.",
        "number_annotators": 6,
        "annotators": ["Annotator_810", "Annotator_811", "Annotator_812", "Annotator_813", "Annotator_814", "Annotator_815"],
        "gender_annotators": ["F", "F", "F", "M", "M", "M"],
        "age_annotators": ["18-22", "23-45", "46+", "46+", "23-45", "18-22"],
        "ethnicities_annotators": ["Hispano or Latino", "White or Caucasian", "White or Latino", "White or Caucasian", "White or Caucasian"],
        "study_levels_annotators": ["High school degree or equivalent", "Master's degree", "Master's degree", "Bachelor's degree", "Bachelor's degree", "Bachelor's degree", "Bachelor's degree"],
        "countries_annotators": ["Mexico", "Spain", "Italy", "United States", "Portugal", "Italy"],
        "split": "TEST_ES"
    },
    "500002": {
        "id_EXIST": "500002",
        "lang": "es",
        "tweet": "@ArcaNgEl_23 @Benzinazi Hombre, no es comparable, mira lo del Gamergate.",
        "number_annotators": 6,
        "annotators": ["Annotator_780", "Annotator_816", "Annotator_817", "Annotator_818", "Annotator_819", "Annotator_820"],
        "gender_annotators": ["F", "F", "M", "M", "M", "M"],
        "age_annotators": ["18-22", "23-45", "46+", "46+", "23-45", "18-22"],
        "ethnicities_annotators": ["Hispano or Latino", "Hispano or Latino", "Black or African American", "Hispano or Latino", "Hispano or Latino", "Hispano or Latino"],
        "study_levels_annotators": ["High school degree or equivalent", "Bachelor's degree", "Bachelor's degree", "Bachelor's degree", "High school degree or equivalent", "Bachelor's degree"]
    }
}
```

Figura 24 Extracto del fichero test proporcionado por la competición

```
[
    {
        "id": "500001",
        "value": {
            "DIRECT": 0.022940158843994002,
            "REPORTED": 0.0,
            "JUDGEMENTAL": 0.022940158843994002,
            "NO": 0.954119682312011
        },
        "test_case": "EXIST2024"
    },
    {
        "id": "500002",
        "value": {
            "DIRECT": 0.07370265324910466,
            "REPORTED": 0.03685132662455233,
            "JUDGEMENTAL": 0.0,
            "NO": 0.8894460201263421
        },
        "test_case": "EXIST2024"
    },
    {
        "id": "500003",
        "value": {
            "REPORTED": 0.04025004705775667
        }
    }
]
```

Figura 25 Fichero de predicciones (de tipo soft label) presentado a la competición

3.12.2 Runs Presentadas para la Task 1

Para el primer par de predicciones (*soft* y *hard*) presentadas para la Task 1, se empleó el modelo clasificador binario llamado v1.1, explicado anteriormente. Este, generó predicciones sobre el archivo de test oficial de la competición, produciendo así dos salidas: task_1_soft_l2C_UHU_1 y task_1_soft_l2C_UHU_1.

Para el segundo par presentado, se usaron los modelos de la estrategia v1.2 para generar dos nuevos archivos de runs de predicciones. El modelo DeBERTa v3 Base predijo las etiquetas de las instancias en inglés y generó un archivo de resultados llamado “task_1_soft_l2C_UHU_2_en”, que incluía las *hard labels* y las *score_label* de la misma. De manera similar, el modelo RoBERTa Base Bne predijo las etiquetas de las instancias en español y generó un archivo de resultados llamado “task_1_soft_l2C_UHU_2_es”, que también incluía las *hard labels* y el *score_label* de la misma.

Posteriormente, se unificaron los resultados de ambos archivos. Los archivos de resultados generados para inglés (“task_1_soft_I2C_UHU_2_en”) y español (“task_1_soft_I2C_UHU_2_es”) se combinaron, generando etiquetas finales tanto en formato de *soft labels* como de *hard labels*. Las *soft labels* se mantuvieron en su estructura original para la predicción conjunta, mientras que se generó un archivo final “task_1_hard_I2C_UHU_2” con las etiquetas mayoritarias (*hard labels*) derivadas de las predicciones combinadas.

3.12.3 Runs Presentadas para la Task 2

Para la Task 2, se presentaron cinco *runs*. Una de tipo *hard label* se generó usando el clasificador de la versión v2.1, produciendo la salida “task_1_hard_I2C_UHU_1”. Dos *runs* adicionales (*hard y soft labels*) se generaron usando el ensemble de los seis modelos de la estrategia v2.2, resultando en los archivos “task_2_soft_I2C_UHU_2” para las *soft labels* y “task_2_hard_I2C_UHU_2” para las *hard labels*, estas últimas basadas en la etiqueta mayoritaria de las *soft labels*. Finalmente, dos *runs* adicionales (*hard y soft labels*) se generaron usando el ensemble de los tres modelos de la estrategia v2.3, produciendo los archivos “task_2_soft_I2C_UHU_3” para las *soft labels* y “task_2_hard_I2C_UHU_3” para las *hard labels*, basadas igualmente en la etiqueta mayoritaria de las *soft labels*.

3.12.4 Resultados Obtenidos

Los resultados obtenidos han sido altamente satisfactorios y demuestran el buen desempeño de las técnicas empleadas en la resolución de ambas tareas. En la primera tarea, *Task 1*, logré alcanzar la sexta posición entre treinta equipos diferentes en la evaluación *hard-hard*. En la evaluación *soft-soft*, obtuve la séptima posición entre diecisiete equipos.

Desglosando el ranking de todas las ejecuciones, obtuve las posiciones 10 y 32 de un total de 70 participaciones en el ranking *hard-hard*. En el ranking *soft-soft*, mis posiciones fueron 13 y 18 de 40 participaciones. Ambos rankings vienen detallados en las Tablas 25 y 26, a continuación.

También quisiera dejar constancia de los buenos resultados obtenidos en el ranking de español para la evaluación *hard-hard* en la *Task 1*. Logré el segundo puesto como equipo y el tercer puesto con una de las ejecuciones presentadas.

Tabla 25 Ranking de participaciones para la evaluación *hard-hard* en la *Task 1*

Run	Rank	ICM-Hard	ICM-Hard Norm	F1_Yes
EXIST2024-test_gold.json	0	0.9948	1.0000	1.0000
NYCU-NLP_1.json	1	0.5973	0.8002	0.7944
...
I2C-UHU_2.json	10	0.5557	0.7793	0.7733
...
I2C-UHU_1.json	32	0.4651	0.7338	0.7513
...
EXIST2024-test_majority-class.json	68	-0.4413	0.2782	0.0000
The-Three-Musketeers_3.json	69	-0.4645	0.2665	0.2999
EXIST2024-test_minority-class.json	70	-0.5742	0.2114	0.5698

Tabla 26 Ranking de participaciones para la evaluación soft-soft en la Task 1

Run	Rank	ICM-Soft	ICM-Soft Norm	Cross Entropy
EXIST2024-test_gold.json	0	3.1182	1.0000	0.5472
NYCU-NLP_1.json	1	1.0944	0.6755	0.9088
...
I2C-UHU_2.json	13	0.6871	0.6102	0.9184
...
I2C-UHU_1.json	18	0.5175	0.5830	1.0666
...
EXIST2024-test_majority-class.json	36	-2.3585	0.1218	4.6115
...
EXIST2024-test_minority-class.json	40	-3.0717	0.0075	5.3572

En la segunda tarea, *Task 2*, logré alcanzar la sexta posición entre treinta y tres equipos distintos en la evaluación hard-hard. En la evaluación soft-soft, obtuve la novena posición entre diecisiete equipos. Desglosando el ranking de todas las ejecuciones, obtuve las posiciones 11, 21 y 24 de un total de 46 participaciones en el ranking hard-hard. En el ranking soft-soft, mis posiciones fueron 17 y 22 de 35 participaciones. Ambos rankings se detallan en las tablas 27 y 28, respectivamente, a continuación.

Tabla 27 Ranking de participaciones para la evaluación hard-hard en la Task 2

Run	Rank	ICM-Hard	ICM-Hard Norm	F1_Yes
EXIST2024-test_gold.json	0	1.5378	1.0000	1.0000
NYCU-NLP_1.json	1	0.4059	0.6320	0.5677
...
I2C-UHU_2.json	11	0.1815	0.5590	0.4980
...
I2C-UHU_1.json	21	0.0418	0.5136	0.4708
...
I2C-UHU_3.json	24	0.0210	0.5068	0.4663
...
EXIST2024-test_majority-class.json	39	-0.9504	0.1910	0.1603
...
EXIST2024-test_minority-class.json	46	-3.1545	0.0000	0.0280

Tabla 28 Ranking de participaciones para la evaluación soft-soft en la Task 2

Run	Rank	ICM-Soft	ICM-Soft Norm	Cross Entropy
EXIST2024-test_gold.json	0	0.9948	1.0000	1.0000
NYCU-NLP_1.json	1	0.5973	0.8002	0.7944
...
I2C-UHU_2.json	17	-2.6952	0.2828	2.1440
...
I2C-UHU_3.json	22	-4.2278	0.1594	2.5245
...
EXIST2024-test_majority-class.json	68	-0.4413	0.2782	0.0000
...
EXIST2024-test_minority-class.json	70	-0.5742	0.2114	0.5698

Como se puede apreciar los mejores resultados han sido logrados por las predicciones generadas por los modelos en los que se aplicaron técnicas de *Learning with Disagreement*, demostrando la efectividad de las técnicas y procedimientos aplicados. Estos resultados serán analizados en el punto siguiente de forma detallada.

CAPÍTULO 4

Conclusiones y Trabajo Futuro

En este apartado se presentan las conclusiones derivadas del estudio realizado, destacando los hallazgos más significativos y su relevancia en el contexto de la investigación. Asimismo, se delinean las posibles líneas de trabajo futuro, identificando áreas que requieren mayor exploración y proponiendo nuevas direcciones que pueden enriquecer el conocimiento en el campo.

4.1 Conclusiones

Tras la experimentación realizada y los resultados oficiales obtenidos en la competición EXIST 2024, se puede concluir que los modelos entrenados bajo el paradigma de Aprendizaje con Desacuerdo han mostrado un rendimiento notable. Este enfoque ha demostrado ser efectivo en la identificación del sexismo y la clasificación de la intención en textos de redes sociales, integrando diversas perspectivas de los anotadores y mejorando así la robustez y precisión de los modelos. La metodología adoptada, que consiste en clasificar los tweets como sexistas o no sexistas y posteriormente categorizar la intención de los tweets sexistas, ha evidenciado mejoras significativas en la comprensión y detección del contenido, en comparación con la clasificación conjunta de todas las etiquetas. En la *Task 1*, se observó un rendimiento superior al emplear modelos específicos de idioma, como RoBERTa Base BNE para español y DeBERTa v3 Base para inglés. Esto resalta la ventaja de utilizar modelos especializados adaptados a cada lengua, optimizando la precisión de la clasificación multilingüe de tweets. Por otro lado, los resultados de la *Task 2* mostraron que el Aprendizaje con Desacuerdo, utilizando un conjunto diverso de anotadores (tanto hombres como mujeres), produjo mejores resultados en comparación con el uso exclusivo de anotadores femeninos. Esto subraya la importancia de la diversidad en la anotación, ya que permite capturar una gama más amplia de matices y sesgos lingüísticos, mejorando así el rendimiento global del modelo. En resumen, el Aprendizaje con Desacuerdo no solo mejora la precisión y robustez de los modelos de clasificación de textos en redes sociales, sino que también promueve una mayor equidad y representatividad en el proceso de anotación.

4.2 Trabajo Futuro

El enfoque futuro del proyecto se centrará en mejorar la precisión y robustez en tareas de identificación y clasificación textual mediante la implementación de modelos de lenguaje de gran escala (LLMs) y la generación de datos sintéticos para mejorar el aprendizaje de estos. Los LLMs han demostrado un rendimiento superior en tareas de procesamiento de lenguaje natural debido a su capacidad para entender y generar texto de manera coherente y contextual. Un ejemplo destacado de LLM para clasificación textual es Mixtral, desarrollado por Mistral AI. Otro modelo relevante es

Falcon, desarrollado por el Technology Innovation Institute (TII) de Abu Dhabi. Falcon ha sido altamente destacado por su capacidad y rendimiento en la comunidad de procesamiento de lenguaje natural. Para abordar el desequilibrio de clases en el conjunto de datos, se utilizará GPT-4 para generar datos sintéticos. Esta estrategia permitirá equilibrar mejor el conjunto de datos creando ejemplos de tweets sexistas y no sexistas que imiten el estilo y contenido de los datos reales. La capacidad de GPT-4 para generar texto en diversos contextos y estilos lingüísticos enriquecerá el conjunto de datos, mejorando la capacidad del modelo para generalizar y su rendimiento en datos no vistos.

4.3 Planificación Temporal del Trabajo Realizado

En la siguiente Tabla 29 se detalla la distribución de horas dedicadas a cada fase crítica del proyecto, garantizando una ejecución eficiente y una presentación detallada de los resultados alcanzados.

Tabla 29 Planificación Temporal del Trabajo Realizado

Tarea o trabajo realizado	Horas
Estudio de la tarea <i>EXIST 2024</i>	5
Aprendizaje de los conceptos necesarios para abordar el proyecto - Conceptos sobre el aprendizaje automático - Redes neuronales - <i>Transformers</i> - Transfer Learning - Fine Tuning - Procesamiento del Lenguaje Natural - Métricas de evaluación de modelos - <i>Learning with Disagreement</i>	60
Participación en el curso “ <i>Desacuerdo y Subjetividad en Aprendizaje Automático: la Importancia del Perspectivismo en el Procesamiento del Lenguaje Natural</i> ”, ofrecido por Simona Frenda (PhD in Computer Science, University of Turin)	6
Participación en seminarios ofrecidos por alumnos de máster y profesores del grupo de investigación I2C	20
Lectura de papers y documentos científicos	15
Aprendizaje de la tecnología necesaria - Python - Pytorch - Plataforma HuggingFace - Optuna - Google Colab - Latex - Modelos de Transfer Learning para clasificación NLP - Jupyter Notebook - Git	80
Realización de la tarea de EXIST 2024 - Descargar y estudiar los datasets - Tratamiento de conjunto de datos - Diseño de experimentos - Implementación del código - Entrenamiento y optimización de los modelos	210
Elaboración de la memoria	50
Elaboración y revisión del paper presentado para EXIST 2024	25
Total	471

Bibliografía

- Amigó, E., & Delgado, A. (2022). *Evaluating Extreme Hierarchical Multi-label Classification*. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics.
- Carrillo-de-Albornoz, J., Amigó, E., Gonzalo, J., Plaza, L., Morante, R., Ruiz, V., . . . Spina, D. (2024). *EXIST 2024: sEXism Identification in Social neTworks and Memes*.
- CLEF Initiative. (2016). CLEF. Obtenido de Conference and Labs of the Evaluation Forum: <https://www.clef-initiative.eu/>
- Conneau, A. (2019). *Unsupervised Cross-lingual Representation Learning at Scale*. arXiv:1911.02116.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., . . . Stoyanov, V. (2020). *Unsupervised Cross-lingual Representation Learning at Scale*. arXiv preprint arXiv:1911.02116.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv preprint arXiv:1810.04805.
- Goldberg, Y. (2015). *A Primer on Neural Network Models for Natural Language Processing*. arXiv:1510.00726.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Grosse, R. (2018). *Multilayer Perceptrons*. University of Toronto.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
- He, P., Liu, X., Gao, J., & Chen, W. (2020). *DeBERTa: Decoding-enhanced BERT with Disentangled Attention*. arXiv preprint arXiv:2006.03654.
- Hugging Face. (2024). *Hugging Face Transformers Tokenizer*. Obtenido de https://huggingface.co/docs/transformers/en/main_classes/tokenizer
- Jurafsky, D., & Martin, J. H. (2024). *Speech and Language Processing. An Introduction to Natural Language Processing (3rd ed.)*. Pearson Prentice Hall.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., . . . Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. arXiv preprint arXiv:1907.11692.
- Ministerio de Educación, Ciencia y Deporte del Gobierno de España. (2018). *Memoria de verificación del Grado en Ingeniería Informática por la Universidad de Huelva*.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Pan, S. J., & Yang, Q. (2010). *A survey on transfer learning*. *IEEE Transactions on knowledge and data engineering*.

Capítulo 4 - Bibliografía

- Rodríguez, R. (2021). *BERT-based models for Spanish*. Technical Report. Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria.
- Russell, S. J., & Norvig, P. (2022). *Artificial Intelligence: A Modern Approach*. Pearson.
- Schmidhuber, J. (2015). *Deep learning in neural networks: An overview*. *Neural Networks*. arXiv:1404.7828.
- Sutton, R. S., & Barto, A. G. (s.f.). *Reinforcement Learning: An Introduction*. 2018: MIT Press.
- Telefónica, E. d. (2020). *Telefónica Tech*. Obtenido de <https://telefonicatech.com/blog/las-matematicas-del-machine-learning-redes-neuronales-ii>
- Tiedemann, J., & Thottingal, S. (2020). *OPUS-MT – Building open translation services for the World*. Lisbon, Portugal: European Association for Machine Translation.
- Uma, A. N., Fornaciari, T., Hovy, D., Paun, S., Plank, B., & Poesio, M. (2021). *Learning from Disagreement: A Survey*; *Dirk Hovy; Silviu Paun; Barbara Plank; Massimo Poesio*. Journal of Artificial Intelligence Research.
- v7labs. (2022). Obtenido de Blog: <https://www.v7labs.com/blog/confusion-matrix-guide>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). *Attention Is All You Need*. arXiv preprint arXiv:1706.03762.
- Vitsakis, N., Parekh, A., Dinkar, T., Abercrombie, G., Konstas, I., & Rieser, V. (2023). *iLab at SemEval-2023 Task 11 Le-Wi-Di: Modelling Disagreement or Modelling Perspectives?* arXiv:2305.06074.

ANEXO I

Repositorios de código en GitHub

En este Anexo I se presenta el repositorio de GitHub donde se encuentra almacenado todo el material necesario para reproducir los experimentos llevados a cabo en el trabajo de fin de grado. En el archivo README incluido en el repositorio se proporcionan instrucciones detalladas sobre cómo acceder y utilizar los recursos disponibles, garantizando así la replicabilidad y transparencia de los procedimientos empleados en el estudio. La documentación contenida en este repositorio facilita la comprensión de la metodología empleada y ofrece la posibilidad de verificar y validar los hallazgos obtenidos en el contexto del trabajo de fin de grado. El repositorio está organizado en dos secciones principales correspondientes a las tareas de la competición EXIST 2024 en las que se participó: *Task 1* y *Task 2*. El enlace de acceso al repositorio es el siguiente:

<https://github.com/mnguerrero11/Learning-from-Divergence-and-Perspectivism-for-Sexism-Identification-Source-Intent-Classification>

Entre los elementos destacados se encuentran:

- **Cuadernos Jupyter Notebook (.ipynb):** Estos archivos contienen el código ejecutable utilizado para llevar a cabo los experimentos, lo que permite una revisión detallada del proceso de análisis y modelado de datos.
- **Conjuntos de datos:** Se incluyen los conjuntos de datos utilizados para entrenar y evaluar los modelos desarrollados en el trabajo. Estos datos son fundamentales para la replicabilidad de los experimentos y la validación de los resultados.
- **Paper de la competición:** Se incluye un documento que describe el flujo de trabajo seguido durante la competición EXIST 2024. Este documento proporciona un contexto más amplio sobre los objetivos, métodos y resultados del estudio, lo que facilita una comprensión completa del trabajo realizado.
- **Código de Python para preprocesamiento de datos y resultados obtenidos:** Además de los cuadernos Jupyter, se proporciona código Python adicional que puede haber sido utilizado para tareas de preprocesamiento de datos, evaluación de modelos y análisis de resultados. Este código complementario contribuye a la comprensión completa del flujo de trabajo empleado en el estudio.

En conjunto, estos elementos constituyen un repositorio completo que brinda a los interesados los recursos necesarios para comprender, replicar y validar los experimentos realizados en el marco del trabajo de fin de grado, promoviendo así la transparencia y reproducibilidad de la investigación.

ANEXO II

Artículo Científico presentado en EXIST 2024

I2C-UHU at EXIST2024: Learning from Divergence and Perspectivism for Sexism Identification and Source Intent Classification

En el marco de la competición EXIST 2024, se requiere que los participantes presenten un artículo explicando los enfoques, metodologías y resultados obtenidos durante la competición. Estos artículos son revisados por pares y forman parte de las actas del CLEF (*Conference and Labs of the Evaluation Forum*). Estos artículos permiten a los investigadores compartir sus hallazgos y contribuciones con la comunidad científica, facilitando la replicación e interpretación de los experimentos realizados.

I2C-UHU at EXIST2024: Learning from Divergence and Perspectivism for Sexism Identification and Source Intent Classification

Manuel Guerrero-García*, Manuel Cerrejón-Naranjo, Jacinto Mata-Vázquez and Victoria Pachón-Álvarez

I2C Research Group, University of Huelva, Spain

Abstract

In this paper, we present the contributions of the I2C-UHU team to the EXIST2024 Lab at CLEF 2024, focusing on the identification of sexism and the classification of source intent in social media texts. State-of-the-art transformer models are employed to address the complex and nuanced nature of sexist language. We adopt a two-fold approach: firstly, classifying tweets as sexist or non-sexist, and secondly, categorizing sexist tweets based on intent. Our innovative approach, employing Learning with Disagreement, incorporates diverse perspectives from multiple annotators, enhancing the robustness and accuracy of our models. We detail our data preprocessing, augmentation techniques, and hyperparameter optimization strategies. Our results in the competition demonstrated effectiveness, with our entries achieving positive rankings in the two tasks in which we participated. In Task 1, we secured the 10th position out of 70 participants on the hard labels leaderboard and the 13th position out of 40 for soft labels. In Task 2, we achieved the 11th position out of 46 participants for hard labels and the 17th position out of 35 in the best run for soft labels. Our findings provide a foundation for future research and practical applications in social media moderation and policy-making.

Keywords

Sexism identification, Learning with disagreement, Transformer models, Natural language processing

1. Introduction

In the EXIST2024 Lab at CLEF 2024[1], the I2C-UHU team addressed sexism on social media platforms through binary classification of tweets and classification based on author intent. The first task distinguishes between sexist and non-sexist content, crucial for filtering harmful language, while the second task classifies sexist tweets into direct, reported, and judgmental categories, providing deeper insights into manifestations of sexism. Utilizing transformer models and data augmentation, our approach aims for robustness and generalizability. By implementing "Learning with Disagreement" [2] we capture diverse perspectives from human annotators, enhancing model accuracy. The paper structure includes sections on related works, dataset description, methodology, results, and future research directions.

2. Related Works

In the realm of detecting sexist tweets, researchers use various methodologies to navigate the complexities of language and intent. Binary classification models serve as a foundational tool, offering a clear distinction between sexist and non-sexist content. However, the quest for a deeper understanding prompts the exploration of author intent, which requires delving into contextual cues and linguistic subtleties.

Task 1 of EXIST 2024 is dedicated to binary categorization, where researchers have explored a spectrum of techniques. From traditional rule-based systems to cutting-edge deep learning architectures, the goal remains consistent: to accurately identify instances of sexism in tweets. Notable among these endeavors is the work of Burnap and Williams [3], who leveraged automatic classification techniques

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France

✉ manuel.guerrero790@alu.uhu.es (M. Guerrero-García); manuel.cerrejon886@alu.uhu.es (M. Cerrejón-Naranjo); mata@uhu.es (J. Mata-Vázquez); vpachon@dti.uhu.es (V. Pachón-Álvarez)

 © 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to detect hate speech on Twitter. Their approach, which incorporated linguistic and contextual features, showcased significant accuracy in pinpointing problematic content.

Task 2, however, takes a deeper dive into the realm of author intent, recognizing that the mere presence of sexist language does not always imply malicious intent. To address this, researchers delve into the intricate interplay between language, context, and underlying motives. Waseem and Hovy [4] embarked on this journey by identifying predictive features for hate speech detection, underscoring the importance of contextual and demographic attributes in discerning the author's intent.

In sum, the exploration of related works underscores the multidimensional nature of detecting sexist tweets. While binary classification models provide a solid foundation, the pursuit of a more nuanced understanding necessitates the integration of author intent analysis and cutting-edge transformer models. These endeavors collectively advance our comprehension of sexism in online discourse and pave the way for more effective mitigation strategies.

3. Tasks and Dataset Description

In this section, the tasks in which participation was engaged and the datasets provided by the organizers are delineated.

3.1. Task 1: Sexism Identification in Tweets

Task 1 involves a binary classification problem where the objective is to determine whether a given tweet contains sexist expressions or behaviors. The classification is straightforward: each tweet is categorized as either sexist ("YES") or not sexist ("NO"). Examples of sexist tweets include statements that directly express sexist sentiments, describe sexist situations, or criticize sexist behaviors. For instance, tweets that demean women's capabilities, perpetuate stereotypes, or contain derogatory comments fall into the "YES" category. Conversely, tweets that do not exhibit these characteristics are labeled as "NO".

3.2. Task 2: Source Intention in Tweets

Task 2 is a multi-class classification task aimed at understanding the intention behind sexist tweets. This task only applies to tweets already identified as sexist in Task 1. The intention of the tweet's author is classified into one of three categories:

- **DIRECT:** The tweet itself is overtly sexist. For example, a tweet stating, "*A woman's place is in the home,*" directly conveys a sexist message.
- **REPORTED:** The tweet reports or describes a sexist incident or situation. An example is, "*Today, I saw a man harass a woman on the subway.*"
- **JUDGEMENTAL:** The tweet condemns or criticizes sexist behaviors or situations. For instance, "*It's disgraceful how women are still paid less than men for the same work.*"

Each of these categories provides insight into the various ways sexism can manifest and the different contexts in which it is discussed on social media.

3.3. Dataset Description

The dataset provided by the organizers contains over 8000 labeled tweets in English and Spanish, with balanced language distribution. The training dataset has 6920 tweets and the development dataset 1038 tweets. Provided in JSON format, each tweet includes attributes such as "id_EXIST", "lang", "tweet", "number_annotation", and detailed annotator information ("annotators", "gender_annotation", "age_annotation", "ethnicity_annotation", "study_level_annotation", "country_annotation"). Labels are "labels_task1" for sexist content and "labels_task2" for author intent. The "split" attribute indicates the dataset subset and language. In Tables 1 and 2 examples of instances for Task 1 and Task 2 are described.

Table 1

Examples of instances for Task 1

id_EXIST	lang	tweet	annotators	labels_task1
101000	es	"No sean de esos que consiguen un peso y cambian con la gente. La plata no es culo."	Annotator_91, Annotator_92, Annotator_93, Annotator_94, Annotator_95, Annotator_96	NO, NO, NO, NO, YES, NO
201573	en	"@Avigeek96 Well men kill women everyday"	Annotator_549, Annotator_550, Annotator_551, Annotator_552, Annotator_553, Annotator_554	NO, YES, YES, YES, YES, YES

Table 2

Examples of instances for Task 2

id_EXIST	lang	tweet	annotators	labels_task2
101000	es	"No sean de esos que consiguen un peso y cambian con la gente. La plata no es culo."	Annotator_91, Annotator_92, Annotator_93, Annotator_94, Annotator_95, Annotator_96	-, -, -, -, JUDGEMENTAL, -
201573	en	"@Avigeek96 Well men kill women everyday"	Annotator_549, Annotator_550, Annotator_551, Annotator_552, Annotator_553, Annotator_554	-, REPORTED, JUDGEMENTAL, JUDGEMENTAL, JUDGEMENTAL, REPORTED

These instances were extracted from the file "*training.json*", which contains 6920 instances, of which 3660 are in Spanish and 3260 are in English. In the case of the file "*dev.json*", which contains 1038 total instances, the language distribution is 549 for Spanish and 489 for English.

In the training and development datasets, the distribution of ethnicities shows a predominant representation of the "White or Caucasian" group, followed by the "Hispanic or Latino" category. Additionally, regarding educational levels, the most common is "Bachelor's degree," while the least represented are "Less than high school diploma" and "Doctorate." The class distributions for both the binary classification task (YES/NO) and the multiclass classification task (DIRECT, REPORTED, JUDGMENTAL) demonstrate substantial consistency between the training and validation datasets. The class distribution in the training and development datasets is depicted in Figure 1.

To effectively apply Learning with Disagreement techniques, it's important to study how different annotator profiles are distributed across the labeled instances.

For training the dataset, there is an equal number of male and female annotators, with a total of 20760 annotators of each gender and a total of 41520. For the development dataset, the distribution is the same, 50% male (3114) and 50% female (3114), out of a total of 6228 annotators. The age distribution is also equitable across both datasets, with one third of annotators falling into each of the following age groups: 18-22 years, 23-45 years, and over 46 years.

4. Methodology

In this section, the methodology used to develop the model submitted to the competition is described.

As previously described, our approaches are based on the use of transformer-based language models. Given that the provided data is in both English and Spanish, four pre-trained models were chosen.

- **XLM-RoBERTa Base:** A pre-trained language model utilizing the RoBERTa architecture and trained on multiple languages. It excels in efficiently and accurately understanding and generating text in various languages[5].

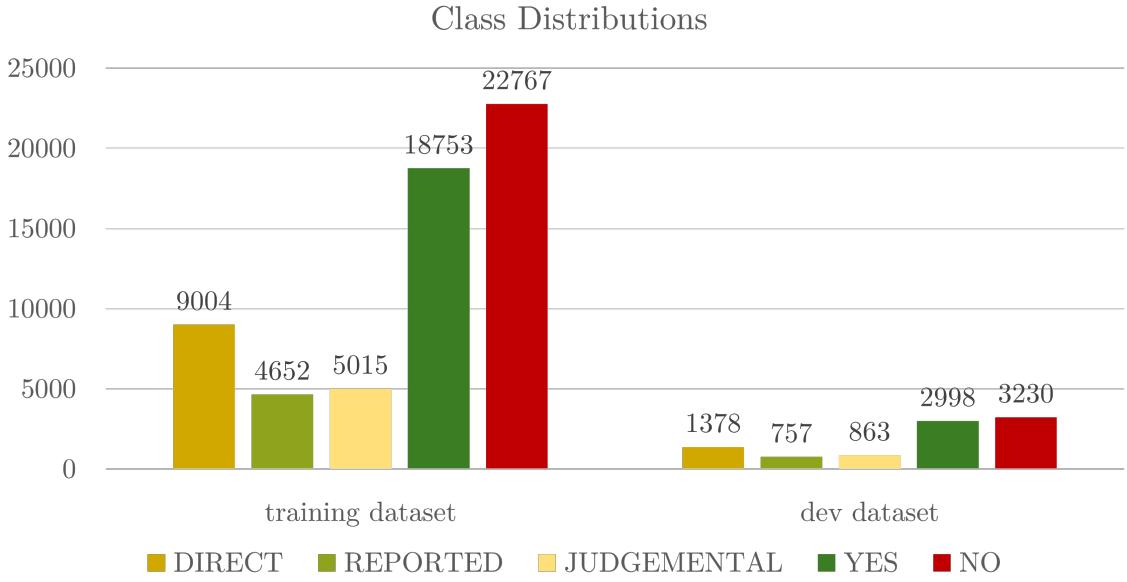


Figure 1: Class distribution in training and dev datasets

- **DeBERTa v3 Base:** A variant of BERT incorporating improvements in attention and word representation, resulting in better performance across a variety of NLP tasks such as text comprehension and language generation[6].
- **RoBERTa Base BNE:** A language-specific adaptation for Spanish of the RoBERTa Base model, trained on the Spanish Corpus from the Spanish Text Bank (BNE). It offers high performance in Spanish language processing tasks[7].
- **BERT Base Multi:** A version of BERT pre-trained in multiple languages and insensitive to case. It can comprehend and generate text in various languages without distinguishing between uppercase and lowercase[8].

4.1. Baseline

The first step in developing classification tasks was to establish an initial benchmark or baseline. This baseline establishes a fundamental methodology that serves as a reference point for comparing more advanced models. It sets a performance threshold that other models must exceed in text classification for our approaches. Two baselines, Version A and Version B, were developed for addressing both Task 1 and Task 2.

4.1.1. Baseline Version A

This approach focuses on training a multiclass classifier (NO, DIRECT, REPORTED, and JUDGEMENTAL) to address all labels for Task 1 and Task 2 simultaneously. The baseline model uses the competition's datasets without preprocessing and with arbitrary hyperparameter values. Both Spanish and English data are included. Models were trained and validated with the training dataset and tested with the development dataset unless otherwise specified for hyperparameter tuning[9].

The hyperparameters values used were: batch size of 32, learning rate of 2e-5, max length of 128, and weight decay of 0.01. The optimizer used was adamw_torch. The maximum number of training epochs was limited to 10 with an "early stopping" set at three epochs.

After training the chosen pre-trained models, the results for the Baseline Version A are presented in Table 3.

This classification strategy yields imprecise and low results. For example, the pre-trained XLM RoBERTa Base model achieved an F1 score of 0.8129 for the NO class, but only 0.3419 for the JUDGE-

Table 3

Results for Baseline Version A

Model	F1 for Baseline A
XLM RoBERTa Base	0.4983
Deberta v3 Base	0.4910
Roberta Base Bne	0.4599
Bert Base Multilingual	0.4388

MENTAL class. This pattern is consistent across other models, indicating difficulty in classifying all the labels together.

4.1.2. Baseline Version B

In Version B, the initial step involves classifying tweets into the two categories of Task 1 (YES and NO). Subsequently, tweets that are categorized as YES are further divided into the three distinct classes of Task 2 (DIRECT, REPORTED, and JUDGEMENTAL). The outcomes achieved with this "Baseline Version B" are detailed in Tables 4 and 5.

Table 4

Results for Baseline Version B, binary classification

Model	F1 for Baseline B
XLM RoBERTa Base	0.7807
Deberta v3 Base	0.7820
Roberta Base Bne	0.7584
Bert Base Multilingual	0.7618

Table 5

Results for Baseline Version B, multiclass classification

Trained Model	F1 for Baseline B
XLM RoBERTa Base	0.568331
Deberta v3 Base	0.555636
Roberta Base Bne	0.530543
Bert Base Multilingual	0.529283

As can be seen, the results improved significantly by breaking down the process into two classification phases.

4.2. Split Description for Training Framework

A schematic overview illustrating the distribution and creation of datasets employed for training the models is shown in Figure 2. These datasets are used in both Task 1 (annotated as v1.x) and Task 2 (annotated as v2.x).

4.3. Data Cleaning and Normalization

In the context of NLP, data cleaning and normalization are critical to ensuring texts are consistent and noise-free before use in machine learning models. Specifically, for cleaning tweets, the following techniques were employed:

- **Lowercase Conversion:** Ensures uniform treatment of words, eliminating the distinction between "Cat" and "cat", simplifying the dataset and reducing the number of unique features.

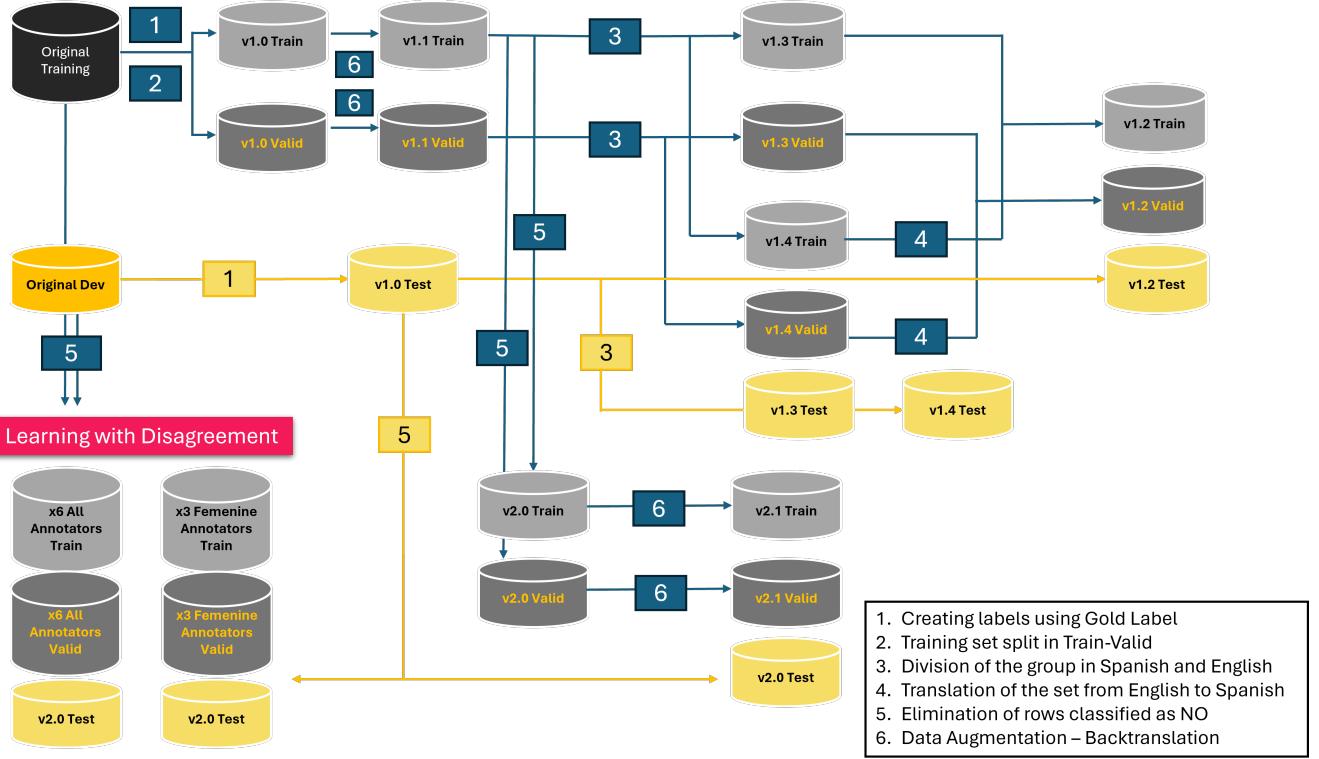


Figure 2: Datasets subdivisions for model training

- **Removal of Links:** Eliminates web links present in tweets as they do not add semantic value and are often irrelevant to sentiment analysis or text meaning.
- **Removal of User Mentions:** Removes mentions of other users and retweets, which usually do not provide relevant information for semantic analysis and can introduce noise.
- **Removal of Hashtags:** Simplifies the text by removing hashtags, which may not be relevant for semantic analysis, focusing the analysis on words and phrases.
- **Removal of Emojis:** Although emojis convey emotions or contexts, their interpretation can be complex in textual analysis. Initial attempts to translate emojis into words did not improve results, thus they were removed to reduce noise and simplify analysis.

An example of the data cleaning carried out is presented in Table 6.

Table 6
Data Cleaning and Normalization

Original Tweet	Cleaned and Normalized Tweet
Collab between WeAreEqual X @TaravaNFT ? YOU ALREADY KNOW IT. Join our Discord on how to join our exclusive Giveaway : https://t.co/x3stzflLmh . #NFT #NFTGiveaway #art	collab between weareequal x ? you already know it. join our discord on how to join our exclusive giveaway : .

4.4. Data Augmentation and Hyperparameter Search

4.4.1. Oversampling with Backtranslation

Oversampling addresses class imbalance [10] by generating syntactic and lexical variations through backtranslation, increasing dataset diversity without altering meaning [11]. Since the datasets are

unbalanced, it is necessary to employ a balancing technique. In this case, the number of rows for the REPORTED and JUDGEMENTAL classes has been increased through backtranslation, while the original number of rows has been maintained for the DIRECT class. Using Helsinki-NLP/opus models from the OPUS project [12], tweets in Spanish are translated to English, then German, and back to Spanish. An example of data generation through backtranslation for a tweet in Spanish is shown in Table 7.

Table 7

Example of data generation through backtranslation for a tweet in Spanish

Original Tweet	New Tweet Generated with Backtranslation
Se supone q me tengo q avergonzar d ser mamá?	¿Debería avergonzarme de ser madre?
Jajajajaajajaja naaaa	

For tweets in English, they were translated from English to German, then from German to Spanish, and finally from Spanish back to English. An example of a newly generated instance is shown in Table 8.

Table 8

Example of data generation through backtranslation for a tweet in English

Original Tweet	New Tweet Generated with Backtranslation
Easy to throw rocks and hide behind your gender or sexual identity #onhere	Easy to throw stones and hide behind your sex or sexual identity #onhere

4.4.2. Hyperparameter Search

Hyperparameter tuning optimizes model performance by selecting optimal values for non-learned parameters. Optuna [13] helps define and iteratively optimize the hyperparameter search space. Exhaustive search (grid search) explores all possible combinations but is computationally expensive. To expedite experiments, the training and validation datasets were reduced to 80% of the original size. To implement exhaustive search using Optuna, a hyperparameter search space was defined, as shown in Table 9.

Table 9

Hyperparameter Search Space

Hyperparameter	Value Range
Batch Size	[8, 16, 32]
Learning Rate	[3e-5, 5e-5]
Weight Decay	[0.001, 0.01, 0.1]

In reference to the metrics obtained after hyperparameter optimization and the application of the previously explained techniques, the results are explained in Tables 10 and 11.

Table 10

F1 scores Task 1

Model	Baseline	Data augmentation + Hyperparameters
XLM RoBERTa Base	0.7807	0.7876
Deberta v3 Base	0.7820	0.7871
Roberta Base Bne	0.7584	0.7616
Bert Base Multilingual	0.7618	0.7640

Table 11
F1 scores Task 2

Model	Baseline	Data augmentation + Hyperparameters
XLM RoBERTa Base	0.5945	0.6095
Roberta Base Bne	0.4795	0.4905
Deberta v3 Base	0.5801	0.5968

4.5. General Training Configuration

Training was conducted using the Trainer class from Hugging Face, incorporating optimized hyperparameters. The adamw_torch[14] optimizer was employed for updating model weights, with evaluations conducted at the end of each epoch and models saved periodically. The best model, determined by the F1 metric, was loaded. Training was halted using the EarlyStoppingCallback if no improvements were observed. These strategies were then tested on the structured dev dataset. The RTX 4070 graphics card was utilized for its high performance and capability to manage intensive processing tasks, thereby ensuring efficient and speedy development and execution of complex models.

4.5.1. Identifying Sexism in Tweets - Version: v1.x

To train the final models that will generate predictions on the test data provided by the competition for Task 1, we selected the two best-performing models based on their metrics during the training process.

To train the v.1.1 model, data from the v1.1 dataset was used. Each tweet in this dataset is labeled by six annotators in both the training and validation sets. To obtain the majority label, following the competition guidelines to obtain the gold label, the votes were averaged, selecting the labels that received two or more votes from among the six possible annotators. In case of a tie, the instance in question was completely excluded. A multilingual model, XLM-RoBERTa-Base, was trained to handle both English and Spanish instances simultaneously. Figure 3 shows this process.

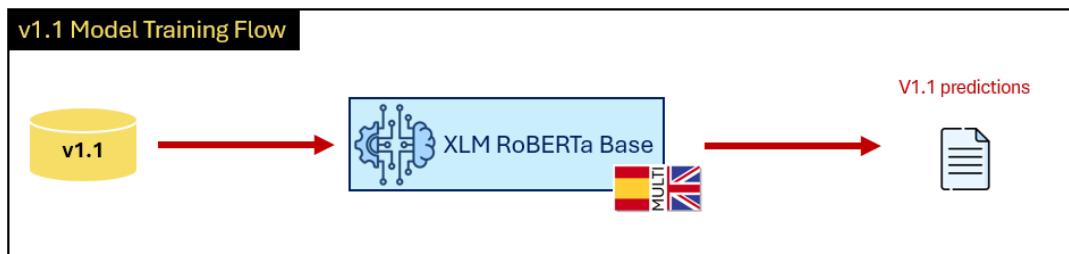


Figure 3: Training flow for model v1.1

Subsequently, this model was used to predict the labels of the data in the official competition test set. The results are presented indicating the majority predicted label for each instance of the test set, followed by the score_label, which represents the similarity score assigned by the classifier to the majority predicted label on a scale of 0 to 1.

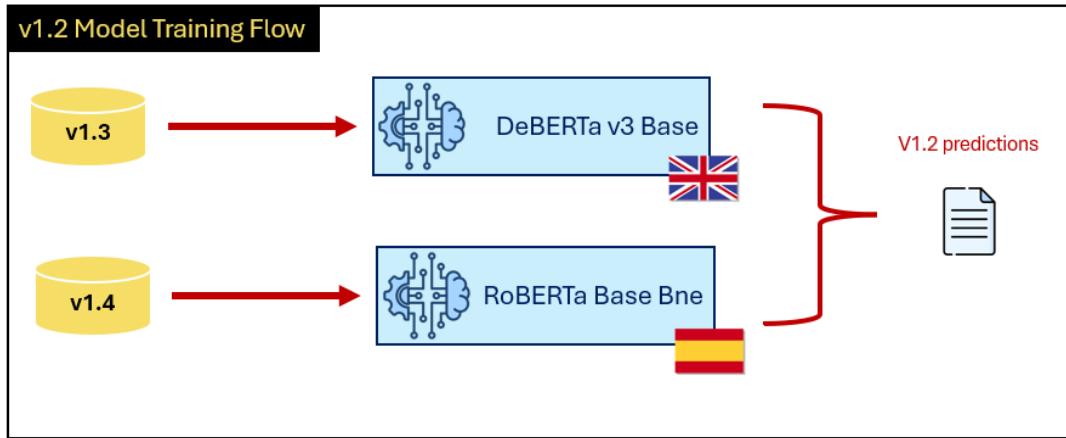
To obtain the hard label, the majority predicted label was selected. Regarding the soft label, since it is a binary classifier (YES or NO), the score_label value was assigned to the majority class in each case, and the value of the minority class was calculated as 1 minus the score_label. It is important to note that the sum of the label values in the soft results should not exceed 1. The results model's evaluation is shown in the Table 12

The training process for model v1.2 is almost identical to the previously described process, but with some key differences regarding the models used and the workflow structure. For training, two datasets were used: v1.3 for English instances and v1.4 for Spanish instances. As for the models, DeBERTa v3 Base was used for English and RoBERTa Base Bne for Spanish. The figure 4 shows the process.

Table 12

Version v1.1 Evaluation models' Results

Model	F1 score
XLM RoBERTa Base	0.850

**Figure 4:** Version v1.2 Evaluation models' Results

The workflow began with the separate training of the two models: the DeBERTa v3 Base model was used for the English instances of dataset v1.3, and the RoBERTa Base Bne model was employed for the Spanish instances of dataset v1.4. The models' evaluation results are shown in Table 13

Table 13

Versions v1.3 and v1.4 Evaluation models' Results

Model	F1 score
XLM RoBERTa Base (v1.3)	0.854
DeBERTa v3 Base (v1.3)	0.859
XLM RoBERTa Base (v1.4)	0.826
RoBERTa Base Bne (v1.4)	0.863
BERT Base (v1.4)	0.818

Table 14

Version v1.2 - Predictions English + Spanish

Model	F1 score
DeBERTa v3 base	0.8589
RoBERTa Base Bne	0.8630
Final Average	0.8617

4.5.2. Intent Classification in Sexist Tweets - Model Versions

Model Version 2.1 was designed to address the second task of the competition, which focuses on classifying the intentionality of tweets previously categorized as sexist by Model Version 1.2 (Source Intention in Tweets). This task follows the initial classification of sexist messages and seeks to categorize such messages according to the author's intent, thus providing insights into the role of social media

in issuing and spreading sexist messages. In this task, a classification between three classes DIRECT, REPORTED, and JUDGEMENTAL is proposed.

The training data comes from Dataset Version 2.1, containing only instances of the three classes, excluding instances categorized as NO, thus avoiding introducing noise in the training data and refining the model's accuracy. Only hard labels were generated for the final predictions, as the model does not return the score label of predicted classes as minority. The Figure 5 shows the process. Obtained results are shown in the Table 15



Figure 5: Training flow for Model Version 2.1

Table 15
Performance of Model Version 2.1

Model	F1 score
XLM RoBERTa Base	0.501

The next model applies Learning with Disagreement because it considers and leverages the differences in opinion among multiple human annotators when labeling the training data. This approach captures a greater diversity of perspectives, which is especially useful in subjective or complex tasks where there may be significant disagreement about the correct labels.

This method improves the model's predictions by integrating multiple viewpoints, creating a more robust and representative training dataset. Additionally, the soft labels resulting from this process enable the model to capture the uncertainty and variability inherent in human annotations, leading to better generalization and performance in real-world situations where data may not be clear or fully defined. The Figure 6 shows the process. Obtained results are shown in the Table 16.

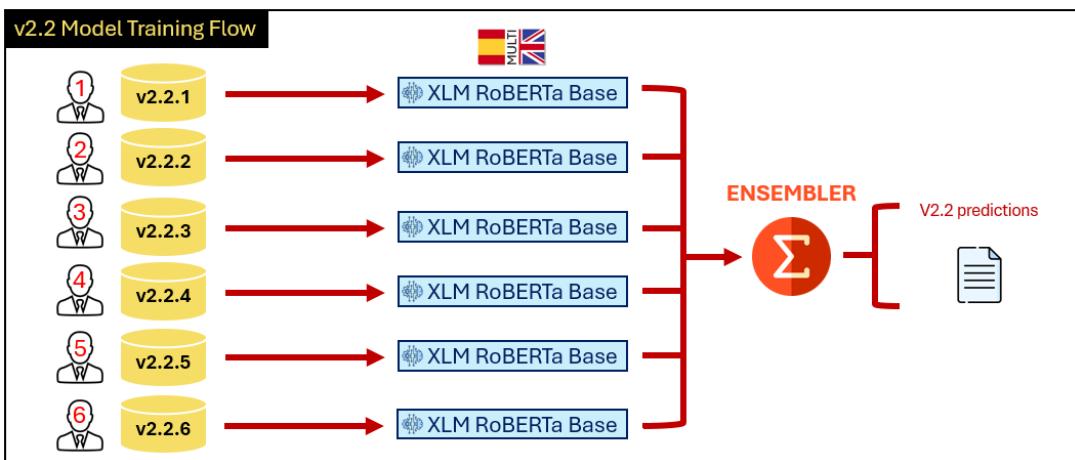


Figure 6: Training flow for Model Version 2.2

The training flow of the model shown in the image can be explained in detail, focusing on how the disagreement among annotators is handled and how soft labels are generated. Here's the step-by-step explanation:

Table 16

Version 2.2 Evaluation models' Results

Model	F1 score
XLM RoBERTa [Ann_1]	0.576
XLM RoBERTa [Ann_2]	0.546
XLM RoBERTa [Ann_3]	0.509
XLM RoBERTa [Ann_4]	0.508
XLM RoBERTa [Ann_5]	0.517
XLM RoBERTa [Ann_6]	0.509
Ensembler	0.527

1. Training data comes from six groups of annotators differentiated by gender and age: ["F 18-22", "F 23-45", "F 46+", "M 46+", "M 23-45", "M 18-22"]. Each group of annotators has provided labels for the training data.
2. Six datasets (v2.2.1, v2.2.2, v2.2.3, v2.2.4, v2.2.5, and v2.2.6) are used to train six instances of the XLM-RoBERTa Base model. Each dataset corresponds to the annotations of one of the six mentioned groups.
3. The six trained models are combined using an ensemble method. This process integrates the outputs of the different models to produce a more robust final prediction. The ensemble calculates a weighted average (sum) of the predictions of the six models.
4. To generate the soft labels, the proportion of annotators who voted for each label is taken into account. For example, if 2 out of 6 annotators labeled a data point as "DIRECT", the soft label for "DIRECT" would be $2/6 = 0.33333$. This process is repeated for the other labels, "REPORTED" and "JUDGEMENTAL".

In the previous task (Task 1), the data was classified into the classes "YES" and "NO". If a data point was classified as "YES" with a probability of 0.80, this value is used to adjust the soft labels of Task 2. For example, if the soft label for "DIRECT" is 0.33333, the adjusted value would be $0.33333 * 0.80 = 0.26666$. This adjustment is performed for all sub-classes of "YES" ("DIRECT", "REPORTED", and "JUDGEMENTAL").

This process must be done for the YES label when it is the majority class in Task 1, as well as to predict the percentage of this when it is the minority class in Task 1. In conclusion, the extremely low probability of the different YES classes in the instances that have been classified by the models of Version 1 as NO is also being calculated.

Finally, Model Version 2.3 follows the same guidelines as Version 2.2, explained above, but the training data comes from three groups of annotators differentiated by gender and age: ["F 18-22", "F 23-45", "F 46+"]. As a sociological experiment, only female groups have been selected to train the models that will compose the ensemble. The Figure 7 shows the process. Obtained results are shown in the Table 18.

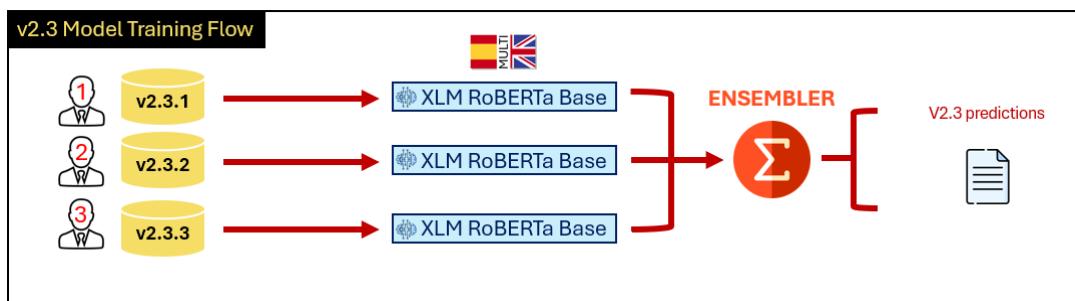
**Figure 7:** Training flow for Model Version 2.3

Table 17
Version 2.3 Evaluation models' Results

Model	F1 score
XLM RoBERTa [Ann_1]	0.575588
XLM RoBERTa [Ann_2]	0.546008
XLM RoBERTa [Ann_3]	0.508656
Ensembler	0.543417

4.6. Error Analysis

4.6.1. Task 1

This section provides a detailed analysis of errors made by the models in Task 1: Sexism Identification in Tweets, focusing on classification discrepancies between YES and NO classes. By scrutinizing misclassifications, patterns and insights into challenges faced by the models are aimed to be identified. Additionally, potential strategies to improve classification performance, especially for the minority class (YES), are explored. Examples are presented in Table 18.

Table 18
Examples of instances for Task 1

Tweet	Labels	Predictions
The worst part is that you also have to be careful about someone seeming too sympathetic and understanding of your struggle, because with the internet comes fetishists and with real life comes the ones who are eager to take advantage of a vulnerable person. It's a sick world.	NO	YES
Woman driving beside me a few minutes ago holding her phone to her ear with her shoulder, while holding a mug of coffee. Baby on Board sticker on both rear windows.	NO	YES
Por qué todos los hombres cuando su novia o esposa está embarazada andan más de culeros que de costumbre.	NO	YES

4.6.2. Task 2

This section analyzes errors encountered by models in Task 2: Source Intention in Tweets, focusing on classification accuracy across DIRECT, REPORTED, and JUDGEMENTAL categories. Through examination of misclassifications, factors influencing performance across these categories are aimed to be understood, and refinements to improve the model's ability to discern nuanced intentions in sexist tweets are discussed. Examples are provided in Table 19, and confusion matrices in Figure 8 depict prediction distributions for Task 2 models.

4.6.3. Error Analysis Conclusions

The analysis of errors in Task 1 and Task 2 uncovers various reasons for misclassifications. Many tweets feature nuanced language or context, challenging for models to interpret. For example, a tweet warning about sympathetic individuals may discuss predatory behavior broadly, misinterpreted by the model as sexist content. Tweets often employ sarcasm, idiomatic expressions, or ambiguous wording, leading to misclassification. A tweet about a woman multitasking while driving may be misconstrued as a gender stereotype critique rather than a comment on unsafe driving practices. Multilingual or culturally referential tweets add complexity. A Spanish tweet discussing men's behavior could be viewed contextually as commentary on male behavior patterns rather than explicit sexism.

Table 19
Examples of instances for Task 2

	Tweet	Labels	Predictions
	Lo irónico es que en su mayoría sean hombres quienes apoyan la criminalización de las mujeres frente al aborto. Claro, a las mujeres hay que castigarlas, juzgarlas y señalarlas siempre, como si no fuera suficiente tener que cargar con el peso de una violación.	REPORTED	DIRECT
	En total delirio esta tipo quiere legalizar el terrorismo. ¿Y esta escoria quiere definir los destinos de Chile? Permitirlo es de anti chilenos.	DIRECT	NO
	If you don't vote, you ARE the problem. #VoteBlueIn2022 #Women-sRights #GunControl #bookban #CivilRights #VotingRights	NO	REPORTED

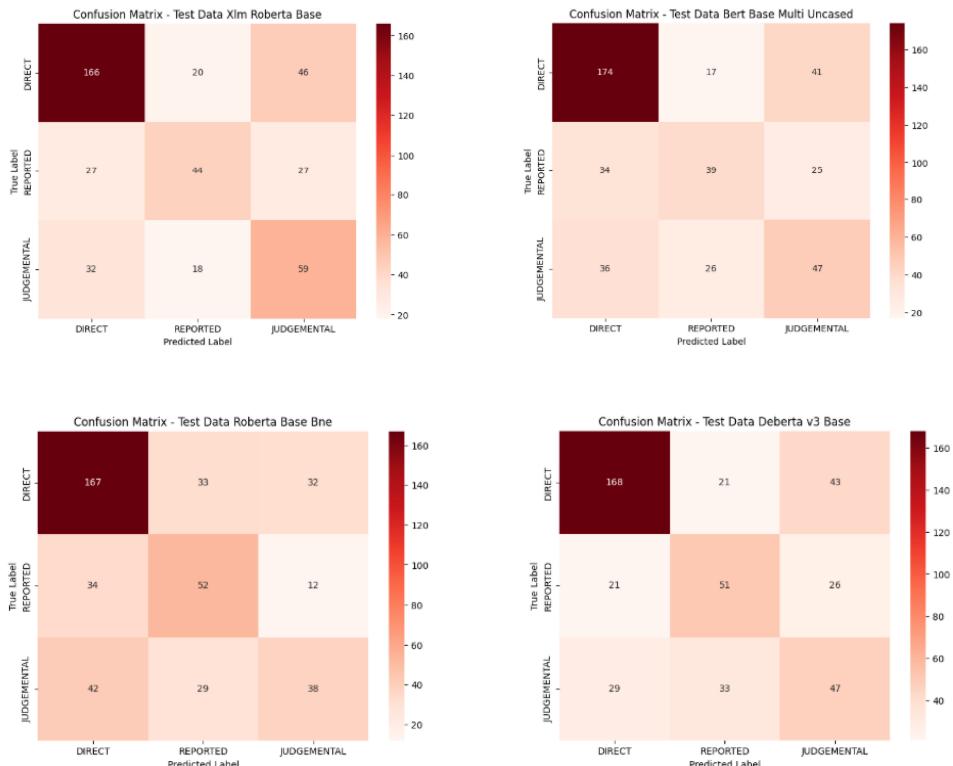


Figure 8: Confusion Matrices for Task 2 models test predictions

5. Oficial Results

In Task 1, the best-performing strategy was a combination of models for different languages: RoBERTa Base BNE was used for classifying Spanish tweets, and DeBERTa v3 Base was employed for English tweets. This dual-model approach significantly outperformed other strategies, emphasizing the effectiveness of leveraging specialized models for each language. Following this, the multilingual model XLM RoBERTa Base also showed strong performance, though it was slightly behind the combined approach. In Task 1, Model v1.1 produced the run I2C-UHU_1, while v1.2 produced I2C-UHU_2. The official results for Task 1 are shown in the tables 20 and 21.

In Task 2, the best results were achieved using the Learning with Disagreement method with six groups of annotators (three male and three female). This approach outperformed the run that applied Learning with Disagreement with only three groups of female annotators. This finding suggests that having a more diverse set of annotators can enhance the model's performance by providing a broader range of perspectives, which likely leads to better generalization and robustness in the model's

Table 20

HARD-HARD Evaluation EXIST 2024 Leaderboard Task1

Ranking	Run	ICM-Hard	ICM-Hard Norm	F1_YES
0	EXIST2024-test_gold.json	0.9948	1.0000	1.0000
-	-	-	-	-
10	I2C-UHU_2.json	0.5557	0.7793	0.7733
-	-	-	-	-
32	I2C-UHU_1.json	0.4651	0.7338	0.7513
-	-	-	-	-
68	EXIST2024-test_majority-class.json	-0.4413	0.2782	0.0000
-	-	-	-	-
70	EXIST2024-test_minority-class.json	-0.5742	0.2114	0.5698

Table 21

SOFT-SOFT Evaluation EXIST 2024 Leaderboard Task1

Ranking	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
0	EXIST2024-test_gold.json	3.1182	1.0000	0.5472
-	-	-	-	-
13	I2C-UHU_2.json	0.6871	0.6102	0.9184
-	-	-	-	-
18	I2C-UHU_1.json	0.5175	0.5830	1.0666
-	-	-	-	-
36	EXIST2024-test_majority-class.json	-2.3585	0.1218	4.6115
-	-	-	-	-
40	EXIST2024-test_minority-class.json	-3.0717	0.0075	5.3572

predictions. For Task 2, v2.1 generated the run I2C-UHU_1, v2.2 produced I2C-UHU_2, and v2.3 resulted in I2C-UHU_3. The official results for Task 2 are shown in the tables 22 and 23.

Table 22

HARD-HARD Evaluation EXIST 2024 Leaderboard Task2

Ranking	Run	ICM-Hard	ICM-Hard Norm	F1_YES
0	EXIST2024-test_gold.json	1.5378	1.0000	1.0000
-	-	-	-	-
11	I2C-UHU_2.json	0.1815	0.5590	0.4980
-	-	-	-	-
21	I2C-UHU_1.json	0.0418	0.5136	0.4708
-	-	-	-	-
24	I2C-UHU_3.json	0.0210	0.5068	0.4663
-	-	-	-	-
39	EXIST2024-test_majority-class.json	-0.9504	0.1910	0.1603
-	-	-	-	-
46	EXIST2024-test_minority-class.json	-3.1545	0.0000	0.0280

Table 23

SOFT-SOFT Evaluation EXIST 2024 Leaderboard Task2

Ranking	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
0	EXIST2024-test_gold.json	3.1182	1.0000	0.5472
-	-	-	-	-
17	I2C-UHU_2.json	-2.6952	0.2828	2.1440
-	-	-	-	-
22	I2C-UHU_1.json	-4.2278	0.1594	2.5245
-	-	-	-	-
27	EXIST2024-test_majority-class.json	-5.4460	0.0612	4.6233
-	-	-	-	-
35	EXIST2024-test_minority-class.json	-32.9552	0.0000	8.8517

6. Conclusions and Future Works

In this paper, the effectiveness of advanced transformer models in addressing the identification of sexism and the classification of source intent in social media texts has been demonstrated. The approach employed, which integrates Learning with Disagreement, facilitates the incorporation of diverse annotator perspectives, thereby enhancing the robustness and accuracy of the models. The methodology, consisting of classifying tweets as sexist or non-sexist and subsequently categorizing the intent of sexist tweets, has shown significant improvements in understanding and detecting nuanced sexist content. The results of the EXIST 2024 Leaderboard for Task 1 and Task 2 provide valuable insights into effective strategies for multilingual tweet classification and the impact of annotator diversity. For Task 1, superior performance was observed with the combination of language-specific models (RoBERTa Base BNE for Spanish and DeBERTa v3 Base for English), indicating the benefit of using specialized models tailored to individual languages. Meanwhile, Task 2 results indicated that Learning with Disagreement, utilizing a diverse set of annotators (both male and female), led to better outcomes compared to using only female annotators. This underscores the importance of diversity in annotation to capture a wider array of linguistic nuances and biases, thus improving the overall performance of the model. Future work will focus on refining the models by incorporating additional data sources and exploring more sophisticated ensemble methods. Additionally, efforts will be made to extend the research to other forms of harmful online content, applying the insights gained from this study to broader applications in social media moderation and policy-making. The insights derived from this research provide a valuable foundation for the development of more effective strategies to combat online sexism and other forms of digital harm.

Acknowledgments

This paper is part of the I+D+i Project titled “*Conspiracy Theories and hate speech online: Comparison of patterns in narratives and social networks about COVID-19, immigrants, refugees and LGBTI people [NON-CONSPIRA-HATE!]*”, PID2021-123983OB-I00, funded by MCIN/AEI/10.13039/501100011033/ and by “ERDF/EU”.

References

- [1] L. Plaza, J. Carrillo-de Albornoz, E. Amigó, J. Gonzalo, R. Morante, P. Rosso, D. Spina, B. Chulvi, A. Maeso, V. Ruiz, Exist 2024: sexism identification in social networks and memes, in: Proceedings of ECIR’24, 2024.

- [2] A. N. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, M. Poesio, Learning from disagreement: A survey, *J. Artif. Int. Res.* 72 (2022) 1385–1470. URL: <https://doi.org/10.1613/jair.1.12752>. doi:10.1613/jair.1.12752.
- [3] P. Burnap, M. L. Williams, Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making, *Policy & Internet* 7 (2015) 223–242. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/poi3.85>. doi:<https://doi.org/10.1002/poi3.85>. arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/poi3.85>.
- [4] Z. Waseem, D. Hovy, Hateful symbols or hateful people? predictive features for hate speech detection on Twitter, in: J. Andreas, E. Choi, A. Lazaridou (Eds.), *Proceedings of the NAACL Student Research Workshop*, Association for Computational Linguistics, San Diego, California, 2016, pp. 88–93. URL: <https://aclanthology.org/N16-2013>. doi:10.18653/v1/N16-2013.
- [5] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, *arXiv preprint arXiv:1906.08237* (2019).
- [6] J. He, Z. Gan, X. Liu, J. Li, J. Gao, Deberta: Decoding-enhanced bert with disentangled attention, *arXiv preprint arXiv:2006.03654* (2021).
- [7] A. Gutiérrez-Fandiño, J. Armengol-Estabé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. Rodriguez-Penagos, M. Villegas, Spanish language models, 2021.
- [8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *arXiv preprint arXiv:1810.04805* (2018).
- [9] T. Yu, H. Zhu, Hyper-parameter optimization: A review of algorithms and applications, 2020. arXiv:[2003.05689](https://arxiv.org/abs/2003.05689).
- [10] S. Kobayashi, Contextual augmentation: Data augmentation by words with paradigmatic relations, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018, pp. 452–457.
- [11] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for nlp, 2021. arXiv:[2105.03075](https://arxiv.org/abs/2105.03075).
- [12] M. Aulamo, J. Tiedemann, The OPUS resource repository: An open package for creating parallel corpora and machine translation services, in: M. Hartmann, B. Plank (Eds.), *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Linköping University Electronic Press, Turku, Finland, 2019, pp. 389–394. URL: <https://aclanthology.org/W19-6146>.
- [13] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. Aji, N. Bogoychev, A. Martins, A. Birch, Marian: Fast neural machine translation in c++, 2018, pp. 116–121. doi:10.18653/v1/P18-4020.
- [14] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. arXiv:[1711.05101](https://arxiv.org/abs/1711.05101).