

**Escuela Técnica Superior de Ingeniería
Universidad de Huelva**

Grado en Ingeniería Informática

Trabajo Fin de Grado

Detección de Sexismo e Intención del Autor en Memes Basado en
Transformers Utilizando un Enfoque de Aprendizaje con Desacuerdo

Álvaro Carrillo Casado
Huelva, Junio 2024

Declaración personal de autoría

Álvaro Carrillo Casado con DNI 49106356E, estudiante del Grado de Ingeniería Informática en la Escuela Superior de Ingeniería de la Universidad de Huelva, como autor de este documento académico titulado Detección de Sexismo e Intención del Autor en Memes Basado en Transformers Utilizando un Enfoque de Aprendizaje con Desacuerdo

DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse. En Huelva, a Junio, 2024

Fdo: Álvaro Carrillo Casado

Agradecimientos

En primer lugar, me gustaría agredecer a mis tutores, Jacinto y Vicky, por impulsarme a realizar este proyecto con todos los retos que ha supuesto que me han ayudado a mejorar.

Gracias a los compañeros y profesores que me han acompañado durante esta etapa, de los que he aprendido día a día.

Por último, a mis padres y mi hermana, los cuales me han apoyado y ayudado en mis mejores y peores momentos.

Álvaro
Huelva, 2024

Resumen

Desde hace un tiempo con el auge de las redes sociales se ha popularizado un tipo de contenido llamado meme, el cual consiste en una imagen o vídeo que contiene una intención humorística, en el caso de la imágenes, donde nos vamos a centrar, se hace normalmente con la inclusión de un texto. Dicho esto también tenemos que tener en cuenta que la sociedad actual trata a veces de usar este contenido humorístico con el objetivo de hacer daño a otras personas, por ejemplo, usando estos memes dotándolos de toques sexistas, de manera clara o utilizando el sarcasmo.

El objetivo del trabajo será analizar la intención con la que se han hecho los memes específicamente de contenido sexistas, para así poder comprender el papel que tienen las redes sociales en la normalización de estas actitudes. Para ello, se han utilizado técnicas de aprendizaje automático y procesamiento del lenguaje natural, apoyados en arquitecturas avanzadas de *Transformers* como son los modelos preentrenados BERT y RoBERTa. Estos modelos son los más utilizados dentro de este tipo de estudios ya que previamente son entrenados para detectar textos tanto de español como de inglés, a los que luego se vuelve a entrenar con los datos centrados en nuestro objetivo.

Se han visto diferentes enfoques a la hora de la creación de los modelos, estudiando tanto las imágenes como el texto que contienen. Mediante el aprendizaje con desacuerdo (*Learning with Disagreement*) se estudiará un enfoque de los datos basado en perspectivas, como pueden ser el sexo, la etnia, el nivel de estudio o la edad, donde cada perspectiva será tratada como un modelo único para luego unir todas las predicciones dándole mayor importancia a las perspectivas que mayor acierto tienen, pero sin perder la opinión de las demás. Se comparará este nuevo enfoque con el estudio clásico de realizar un único modelo.

Finalmente, para obtener los datos y una manera de evaluar y comparar el trabajo realizado se ha participado en la 4º edición de 'Identificación de Sexismo en Redes Sociales' de EXIST 2024 de la 15º edición de *Conference and Labs of the Evaluation Forum (CLEF 2024)*, más concretamente en la 4º y 5º tarea. En esta tendremos que estudiar previamente los memes para determinar si son sexistas o no, para luego tratar si tienen una intención directa o sarcástica. Por último, el estudio se concluirá con la publicación de un paper científico.

Palabras clave: Memes, sexismo, intención, Transformers, aprendizaje con desacuerdo, hiperparámetros.

Abstract

For some time now, with the rise of social media, a type of content known as memes has become popular. A meme typically consists of an image or video intended to be humorous. In the case of images, which we will focus on, this often involves the inclusion of text. It's important to note that contemporary society sometimes employs such humorous content with the aim of harming others, for instance, by imbuing these memes with sexist undertones, either explicitly or through the use of sarcasm.

The objective of this study is to analyze the intent behind specifically sexist memes, aiming to understand the role that social media plays in normalizing these attitudes. To achieve this, machine learning techniques and natural language processing have been employed, leveraging advanced Transformer architectures such as the pretrained models BERT and RoBERTa. These models are widely used in such studies as they are trained to detect texts in both Spanish and English, and subsequently fine-tuned with data focused on our objectives.

Various approaches have been observed in model creation, examining both the images and the textual content they contain. Using the Learning with Disagreement method, an approach based on different perspectives such as gender, ethnicity, education level, or age will be studied. Each perspective is treated as a unique model, and predictions from these models are aggregated, giving more weight to perspectives that are more accurate while still considering others' opinions. This new approach will be compared with the traditional method of creating a single model.

Finally, to gather data and evaluate/compare the work done, participation occurred in the 4th edition of Identifying Sexism in Social Media.^at EXIST 2024, part of the 15th edition of the Conference and Labs of the Evaluation Forum (CLEF 2024), specifically in tasks 4 and 5. The study involves pre-analyzing memes to determine their sexist nature and whether their intent is direct or sarcastic. Ultimately, the research will conclude with the publication of a scientific paper.

Keywords: Memes, sexism, Intention, Transformers, Learning with Disagreement, Hyperparameters.

Índice general

Índice de figuras	XI
Índice de tablas	XIII
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	1
1.3. Competencias Adquiridas	1
1.4. Estructura de la Memoria	2
2. Marco Teórico	3
2.1. Aprendizaje Automático	3
2.2. Aprendizaje Profundo (<i>Deep Learning</i>)	3
2.3. Visión por Computador	4
2.4. Procesamiento del Lenguaje Natural (PLN)	5
2.5. Transformers	6
2.5.1. BERT	7
2.5.2. RoBERTa	8
2.5.3. BEiT	8
2.5.4. ViT	8
2.6. Aprendizaje por Transferencia (<i>Transfer Learning</i>)	9
2.7. Medidas de Evaluación	9
2.8. Ensemble de Modelos	11
2.9. Aprendizaje con Desacuerdo (<i>Learning with Disagreement</i>)	11
2.10. Tecnología y Recursos Utilizados	11
3. Metodología, Experimentación y Resultados	13
3.1. Tarea	13
3.2. Descripción de los Datos	14
3.3. Metodología	16
3.4. Estudio de Imágenes y Textos	17
3.4.1. Selección de Modelos	17
3.4.2. Baseline	18
3.4.3. Análisis del Estudio	19
3.5. Preprocesamiento de Datos	19
3.5.1. Limpieza de Datos	19
3.5.2. Tokenización y Codificación	20
3.6. Ajuste de Hiperparámetros	21
3.7. Enfoque Aprendizaje con Desacuerdo	22
3.8. Construcción de los Datasets	23
3.9. Entrenamiento de los Modelos	24
3.10. Validación y Evaluación de los Modelos	25
3.10.1. Evaluación de Perspectivas	25
3.10.2. Ensemble de Modelos	25
3.11. Análisis de Errores	27
3.12. Resultados Oficiales de la Competición	29

4. Conclusiones y Trabajo Futuro	31
4.1. Conclusiones	31
4.2. Trabajo Futuro	31
4.3. Planificación Temporal del Trabajo Realizado	32
Bibliografía	33
Anexos	34

Índice de figuras

2.1. Estructura <i>Deep Learning</i>	4
2.2. Proceso de visión por computador	5
2.3. Funcionamiento de una Red Convolucional (CNN)	5
2.4. Ejemplo de tokenización	6
2.5. Arquitectura <i>Transformers</i>	7
2.6. Esquema del proceso de BERT	8
2.7. Esquema de funcionamiento <i>Transfer Learning</i> vs <i>Fine-tuning</i>	9
2.8. Matriz de confusión	10
3.1. Ejemplo memes tarea 4	14
3.2. Ejemplo memes tarea 5	14
3.3. Ejemplo de características para cada meme	15
3.4. Metodología del proyecto	17
3.5. Ejemplo memes sexistas diferentes tarea 4	19
3.6. Ejemplo de entrada al proceso de tokenización y codificación	20
3.7. Ejemplo de salida del proceso de tokenización y codificación	21
3.8. <i>Max Length</i> para tokenización	21
3.9. Ejemplo visualización durante el entrenamiento	24
3.10. Matrices de confusión para la tarea 4	27
3.11. Matrices de confusión para la tarea 5	28

Índice de tablas

3.1.	Distribución de clases para la tarea 4	16
3.2.	Destribución de clases para la tarea 5	16
3.3.	Baselines de modelos imágenes para tarea 4	18
3.4.	Baselines de modelos texto para tarea 4	18
3.5.	Baselines de modelos imagenes para tarea 5	18
3.6.	Baselines de modelos texto para tarea 5	19
3.7.	Limpieza de datos para la tarea 4 (F1-Score)	20
3.8.	Limpieza de datos para la tarea 5 (F1-Score)	20
3.9.	Espacio de hiperparámetros	21
3.10.	Mejores hiperparámetros para la tarea 4	22
3.11.	Mejores hiperparámetros para la tarea 5	22
3.12.	Balanceo de datos de las perspectivas de la tarea 4	23
3.13.	Balanceo de datos de las perspectivas de la tarea 5	23
3.14.	Resultados F1-Score para cada perspectiva de la tarea 4	25
3.15.	Resultados F1-Score para cada perspectiva de la tarea 5	25
3.16.	Espacio de pesos para cada predicción individual	26
3.17.	Mejores combinaciones para la tarea 4	26
3.18.	Mejores combinaciones para la tarea 5	26
3.19.	Ejemplos de etiquetado tarea 4	28
3.20.	Ejemplos de etiquetado para la tarea 5	29
3.21.	Ranking de participantes para la tarea 4 Hard-Hard	29
3.22.	Ranking de participantes para la tarea Soft-Soft	30
3.23.	Ranking de participantes para la tarea 5 Hard-Hard	30
3.24.	Ranking de participantes para la tarea 5 Soft-Soft	30
4.1.	Planificación Temporal del Trabajo	32

1. Introducción

En este capítulo, se describe la motivación y objetivos para la realización de este proyecto. Se explicarán las competencias adquiridas y se introducirá brevemente la estructura de la memoria.

1.1. Motivación

Los memes se han convertido en una parte omnipresente de la cultura digital, actuando como vehículos para la transmisión de ideas, opiniones y, a menudo, estereotipos. Por ello, la detección de sexismo y la intención del autor detrás de los memes es crucial en un mundo cada vez más digitalizado, pudiendo influir en la percepción y comunicación social de las personas. Para contribuir en la lucha por promover la igualdad de género surge la necesidad de desarrollar técnicas de procesamiento de lenguaje natural (PLN) que permitan resolver este problema.

1.2. Objetivos

El objetivo principal de este proyecto es obtener conocimientos de aprendizaje automático y *Deep Learning* para detectar sexismo y comprender la intención del autor en memes. Para conseguir este objetivo se buscará lograr los siguientes objetivos individuales:

- Estudiar diferentes técnicas de *Deep Learning* para la resolución de la tarea.
- Examinar las imágenes como enfoque para resolver el problema.
- Analizar los textos como medio para resolver el problema.
- Estudiar los modelos mas eficaces para cada tipo de dato.
- Diseñar mejoras para la optimización de los modelos.
- Mejorar el funcionamiento de los modelos explorando distintas técnicas.

Para la evaluación de los resultados obtenidos se ha participado en '*EXIST: sEXism Identification in Social neTworks*'¹, específicamente en las tareas: '*Task 4: Sexism Identification in Memes*' y '*Task 5: Source Intention in Memes*'. Finalmente, se realizó un artículo científico para describir la metodología empleada.

1.3. Competencias Adquiridas

La principal competencia adquirida ha sido ÇEC-7 Capacidad para conocer y desarrollar técnicas de aprendizaje computacional y diseñar e implementar aplicaciones y sistemas que las utilicen, incluyendo las dedicadas a extracción automática de información y conocimiento a partir de grandes volúmenes de datos.

¹<http://nlp.uned.es/exist2024/>

Otras de las competencias adquiridas durante la realización del proyecto han sido:

- Capacidad para conocer y desarrollar técnicas de aprendizaje computacional.
- Capacidad para implementar aplicaciones y sistemas que la utilicen.
- Capacidad para la extracción automática de información.
- Capacidad para adquirir conocimiento a partir de grandes volúmenes de datos.
- Capacidad para la creación y adaptación de modelos basados en *Transformers*.
- Capacidad para estudiar y ajustar hiperparámetros mediante el uso de bibliotecas de *Python*.
- Capacidad para evaluar la efectividad de los modelos desarrollados mediante el uso de métricas de rendimiento.
- Capacidad para utilizar *GitHub* como plataforma en la nube para la gestión de repositorios de códigos.
- Capacidad para la realización de documentos científicos.

1.4. Estructura de la Memoria

El resto de la memoria se organiza de la siguiente manera:

- En el capítulo 2, Marco Teórico, se presentan los fundamentos del aprendizaje automático y *Deep Learning*, así como las arquitecturas de los modelos empleados para el estudio de las imágenes y los textos.
- En el capítulo 3, se expone las tareas propuestas por la competición y la metodología empleada para su resolución.
- En el capítulo 4, Propuestas de Mejora, se plantean nuevas formas para la optimización de los resultados obtenidos en la competición.
- En el capítulo 5, Conclusiones y Trabajo Futuro, se exponen las conclusiones de este proyecto y se proponen posibles direcciones para futuras investigaciones.
- Finalmente, en el Anexo, se incluye el repositorio del código en *GitHub* y el artículo científico presentado a la organización.

2. Marco Teórico

En esta sección se exponen los fundamentos teóricos necesarios para llevar a cabo el proyecto. Se describen los principios básicos del aprendizaje automático y del aprendizaje profundo (*Deep Learning*), procesamiento del lenguaje natural (PLN), *Transformers*, además de conceptos como el aprendizaje por transeferencia y el aprendizaje con desacuerdo (*Learning with Disagreement*) entre otros.

2.1. Aprendizaje Automático

El aprendizaje automático (*Machine Learning*) es una rama de la inteligencia artificial que permite a los sistemas aprender de los datos y mejorar su rendimiento a la hora de tomar decisiones [1].

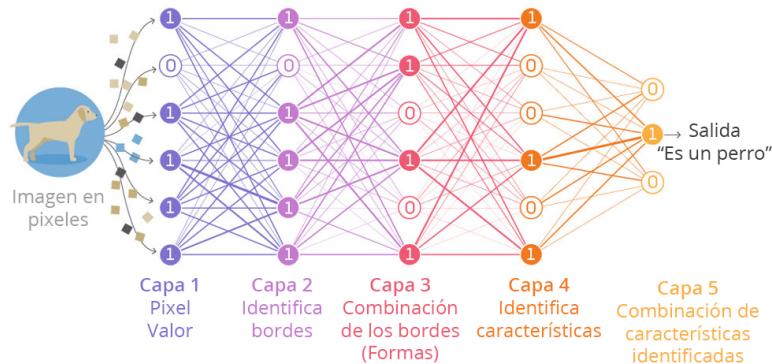
Este aprendizaje permite realizar tareas específicas de manera autónoma sin la necesidad de programación. Se encuentran tres tipos de aprendizaje:

- **Aprendizaje Supervisado** [2]. El aprendizaje supervisado es una técnica utilizada en el aprendizaje automático donde el conjunto de datos de entrenamiento está previamente etiquetado. Estos se utilizan normalmente para la clasificación de unas instancias diferentes a las utilizadas para el entrenamiento. Un ejemplo de ello es la detección del sexismo en tweets, donde dado un texto (tweet) el algoritmo es capaz de detectar si el tweet contiene contenido sexista o no [3].
- **Aprendizaje No Supervisado** [2]. El aprendizaje no supervisado se define como un modelo predictivo entrenado de una forma similar a la del aprendizaje supervisado pero los datos usados para el entrenamiento no están etiquetados, por lo que trata de buscar patrones similares en los datos para agruparlos.
- **Aprendizaje por Refuerzo** [4]. El aprendizaje por refuerzo, se basa en montar un sistema de recompensas en vez de los datos, es decir, el sistema realiza acciones y en caso de que estas sean favorables para la resolución del objetivo recibe una recompensa positiva. Sin embargo, si es desfavorable la recompensa será negativa. El objetivo final de este aprendizaje es maximizar la puntuación de las recompensas tras conseguir resolver el problema.

Todos tienen sus ventajas y desventajas dependiendo de la meta a resolver por el sistema. En este proyecto, se ha utilizado el aprendizaje supervisado ya que el objetivo es clasificar memes etiquetados por diferentes anotadores.

2.2. Aprendizaje Profundo (*Deep Learning*)

El aprendizaje profundo (*Deep Learning*) [5], es una rama del aprendizaje automático que se inspira en la estructura y el funcionamiento de la red neuronal de un cerebro humano. Se utilizan múltiples capas para aprender grandes volúmenes de datos, ya que a diferencia de los métodos tradicionales del aprendizaje automático, la fase de aprendizaje está automatizada respecto a la selección de características y procesos de aprendizaje del *Machine Learning*. Esto facilita el rendimiento de los clasificadores en tareas complejas.

Figura 2.1: Estructura *Deep Learning*

El desarrollo de esta técnica ha sido impulsado por la disponibilidad de grandes volúmenes de datos, lo que permite el entrenamiento de las redes neuronales profundas. A día de hoy las redes neuronales superan a los aprendizajes convencionales en varios dominios como son la visión por computador y el procesamiento del lenguaje natural.

El entrenamiento de una red neuronal incluye: inicializar pesos aleatoriamente, realizar la propagación hacia adelante para obtener predicciones, calcular la pérdida entre las predicciones y las etiquetas reales, aplicar retropropagación para ajustar los pesos utilizando los gradientes de la pérdida, y actualizar los pesos iterativamente mediante optimizadores como Adam o SGD. Este proceso se repite a través de múltiples épocas para mejorar la eficiencia y los resultados.

El aprendizaje profundo continúa mejorando en áreas como la eficiencia y la precisión de los modelos. Se encuentran en estudios de nuevas arquitecturas como son las redes convolucionales (CNN) [6] para el procesamiento y análisis de imágenes. Las redes recurrentes (RNN) [7] se usan para el procesamiento de secuencias de datos especialmente para tareas de orden o dependencia temporal.

2.3. Visión por Computador

La visión por computador [8], es una técnica centrada en la interpretación y análisis de las imágenes y videos. Su objetivo principal es que los computadores comprendan y procesen imágenes de manera similar como lo hace el humano, esto incluye tareas como detección de objetos, clasificación de imágenes y reconocimiento de patrones. Se han visto grandes avances en esta rama gracias a las redes neuronales convolucionales (CNN), ya que son capaces de extraer características complejas y aprender de ellas.

Componentes del proceso:

- **Preprocesamiento de imágenes:** escalado, normalización y reducción de ruido.
- **Extracción de características:** uso de técnicas como redes neuronales convolucionales (CNN).
- **Análisis y clasificación:** imágenes procesadas se analizan para clasificar objetos, detectar errores...
- **Aplicaciones:** se utiliza en áreas como la medicina, vehículos, seguridad...

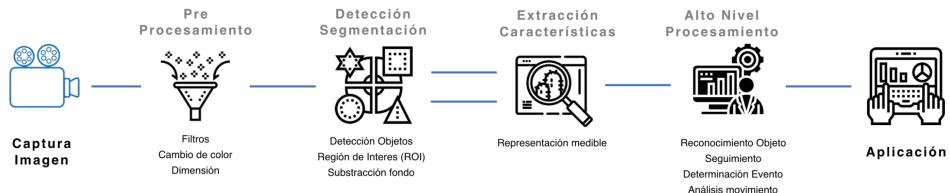


Figura 2.2: Proceso de visión por computador

En las redes convolucionales (CNN), diferenciamos dos conceptos importantes, uno es el *Padding*, el cual indica cuánto borde se le coloca a las imágenes. El otro es *Pooling* [9], el cual se encarga de reducir las dimensiones de las imágenes, reduciendo el número de píxeles de la matriz final. Entre los diferentes tipos de *Pooling*, todos dividen las imágenes en regiones y se encuentran las siguientes opciones:

- **Max Pooling:** toma el mayor valor de la región.
- **Average Pooling:** toma el valor promedio de la región.
- **Global Average Pooling:** toma el valor promedio de todas las activaciones características.
- **Sum Pooling:** suma los valores de los píxeles.
- **Min Pooling:** toma el valor mínimo de cada región.

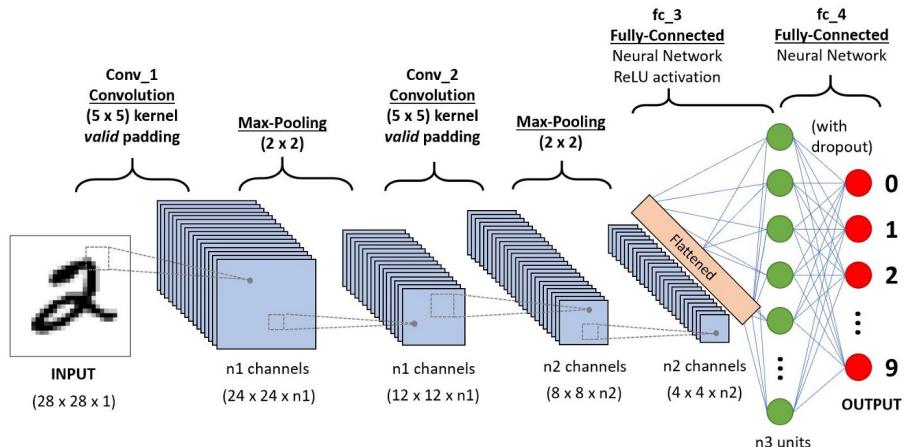


Figura 2.3: Funcionamiento de una Red Convolutinal (CNN)

2.4. Procesamiento del Lenguaje Natural (PLN)

El procesamiento del lenguaje natural [10], trata de desarrollar modelos que permitan a los computadores entender, interpretar y generar texto como los humanos. Algunos de sus componentes fundamentales son la tokenización, donde se divide el texto en pequeñas unidades para facilitar el proceso, y la normalización, mediante la cual se eliminan caracteres especiales y se convierte de mayúsculas a minúsculas para estandarizar el texto.

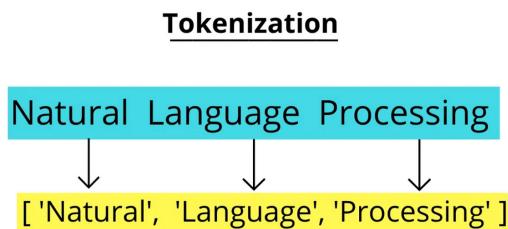


Figura 2.4: Ejemplo de tokenización

Para la representación del lenguaje se usan dos técnicas:

- ***Bag of Words***: representa el texto como un conjunto de palabras sin tener en cuenta el orden.
- ***Word embeddings***: vectores densos para capturar el significado semántico de las palabras.

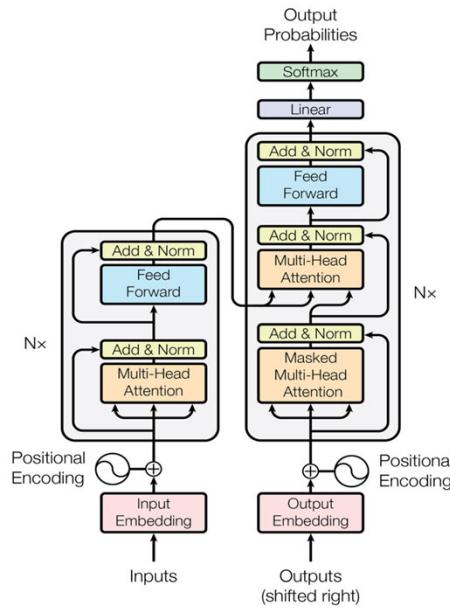
En este tipo de problemas, suelen ocurrir durante la experimentación un desbalanceo de los datos [11], producido porque alguna clase cuenta con más datos que otras. Para ello, aplicamos diferentes técnicas de balanceo:

- ***OverSampling***: aumento de número de instancias de la clase minoritaria generando muestras sintéticas de las ya existentes.
- ***UnderSampling***: reducir el número de intancias de la clase mayoritaria al tamaño de la clase minoritaria.

El PLN ha sufrido un gran avance gracias al *Deep Learning* y al uso de modelos basados en *Transformers*, mejorando la capacidad de compresión y generación de lenguaje natural de los computadores.

2.5. Transformers

Los *Transformers* [10], son modelos basados en mecanismos de atención sin utilizar redes recurrentes ni convolucionales. Utiliza una capa de atención multi-cabeza para capturar dependencias de largo alcance y relaciones semánticas entre palabras en una oración, lo que hace que esta técnica supere a los modelos tradicionales en velocidad de entrenamiento y precisión en diferentes tareas.

Figura 2.5: Arquitectura *Transformers*

Como se puede ver en la Figura 2.5, los *Transformers* están divididos en dos partes. La primera es un codificador, encargado de leer y procesar la entrada, mientras que la segunda es un decodificador, el cual genera la salida.

El uso de este tipo de arquitectura ha provocado que se puedan capturar dependencias a largo plazo gracias a su mecanismo de atención, lo que ayuda a mejorar la calidad de las traducciones. También trabaja mejor con conjuntos de datos grandes y pueden capturar el contexto global de las palabras por lo que generan representaciones vectoriales más contextualizadas.

En el funcionamiento se introducen conceptos como *Padding*, que es el proceso de añadir tokens a secuencias más cortas de datos para hacerlas de igual tamaño que las más largas. Estas son definidas por el *Max Length* (Tamaño máximo de secuencia), permitiendo que el modelo maneje de manera más eficiente secuencias de longitud variable durante el entrenamiento. Para ello se aplica una máscara de atención, la cual hace que el modelo no procese estos tokens.

Los *Transformers* han significado un gran avance en el procesamiento del lenguaje natural, mejorando tareas como traducción automática de textos, análisis de sentimientos y generación de textos. También han demostrado ser efectivos en tareas de visión por computador como la clasificación de imágenes, segmentación semántica y generación de descripciones de imágenes.

2.5.1. BERT

BERT (*Bidirectional Encoder Representations from Transformers*) [12], desarrollado por Google, es una técnica basada en redes neuronales para el procesamiento del lenguaje natural. Su mecanismo se basa en aprender las relaciones contextuales entre palabras, de manera bidireccional, es decir, teniendo en cuenta tanto el contexto de las palabras que lo preceden, como las que lo siguen. Incluye un codificador que lee el texto y un decodificador que produce la predicción.

Utiliza dos técnicas como son Máscara de Lenguaje (MLM), para predecir palabras enmascaradas dentro de una oración, y Predicción de la Próxima Oración (NSP), para determinar si la oración sigue lógicamente a la otra.

La entrada del codificador es una secuencia de tokens, que primero se convierten en vectores para luego procesarse en la red neuronal. En la Figura 2.6, se puede apreciar el proceso de funcionamiento de BERT, donde se añade un token a los tokens de las palabras al principio de la primera frase y otro al final, *Token Embeddings*. Luego se añade un marcador a cada token, *Segment Embeddings*, para finalmente, incluir una incrustación posicional a cada token, *Posicional Embeddings*.

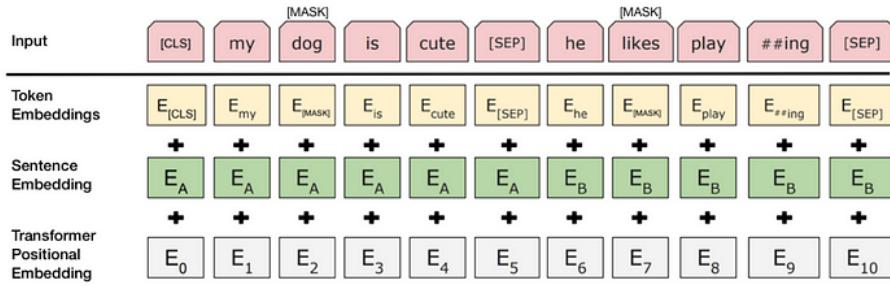


Figura 2.6: Esquema del proceso de BERT

2.5.2. RoBERTa

RoBERTa (*Robustly Optimized BERT approach*) [13], desarrollado por Facebook, es una optimización de BERT entrenado con 10 mil datos más. A diferencia de BERT, elimina la predicción de la siguiente oración e introduce un enmascaramiento dinámico para que los tokens cambien durante el entrenamiento.

Una de las novedades de RoBERTa fue entrenar con gran cantidad de datos sin etiquetar para poder aprender representaciones más generales.

2.5.3. BEiT

BEiT (*Bidirectional Encoder representation from Image Transformers*) [14], es un modelo de visión por computador que utiliza *Transformers* como su inspiración BERT. Preentrena los modelos en tareas de reconstrucción de imágenes, similar al enfoque de enmascaramiento de palabras de BERT, siendo también bidireccional.

Esta especializado en tareas de visión y suele tener un rendimiento superior a los demás ya que comprende de manera profunda las relaciones globales en las imágenes.

2.5.4. ViT

ViT (*Vision Transformers*) [15], es una arquitectura de redes neuronales basada en *Transformers* diseñada originalmente para procesamiento del lenguaje natural. Se estudió para tareas de visión por computador, para abordar las limitaciones de las redes convolucionales (CNN). Las imágenes se dividen en píxeles de 16x16 convirtiéndose en vectores de características y enriqueciéndose del uso de embeddings.

Esta arquitectura supera el rendimiento de las redes convolucionales cuando se usan grandes cantidades de datos, pero computacionalmente es más costoso.

2.6. Aprendizaje por Transferencia (*Transfer Learning*)

El aprendizaje por transferencia (*Transfer Learning*) [16], es una técnica muy utilizada en el mundo de las redes neuronales. Se selecciona una arquitectura ya preentrenada, normalmente relacionada con el objetivo del estudio, y se le añaden los datos correspondientes. En esta cabe destacar que solo se modifican los pesos de las capas añadidas por el desarrollador.

Una de las técnicas más utilizadas en este ámbito es el *Fine-tuning* [17], la cual es más compleja porque también entrena los pesos del modelo ya preentrenado calculando su gradiente.

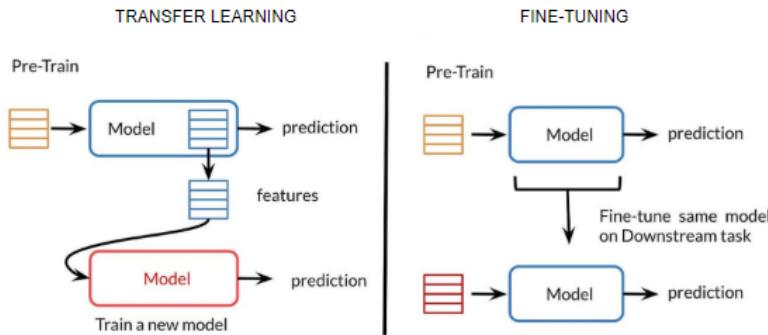


Figura 2.7: Esquema de funcionamiento *Transfer Learning* vs *Fine-tuning*

2.7. Medidas de Evaluación

Durante la experimentación de un proyecto una de las herramientas fundamentales son las medidas de evaluación [18], estas permiten cuantificar el rendimiento y precisión de los modelos. Para entender como funcionan las diferentes medidas se tiene que introducir el concepto de matriz de confusión, siendo esta una representación matricial de los resultados de las predicciones.

Una predicción puede tener los siguientes resultados:

- **Verdadero Positivo (TP):** predicho verdadero y resultado real verdadero.
- **Verdadero Negativo (TN):** predicho falso y resultado real falso.
- **Falso Positivo (FP):** predicho verdadero y resultado real falso.
- **Falso Negativo (FN):** predicho falso y resultado real verdadero.

En la Figura 2.8, se puede ver el resultado de la matriz de confusión, situándose en la diagonal de izquierda a derecha los aciertos del modelo y de derecha a izquierda los errores.

		Actual Values	
		Yes	No
Predicted Values	Yes	True Positive	False Positive
	No	False Negative	True Negative

Figura 2.8: Matriz de confusión

Existen distintas métricas de evaluación pero las más importantes son las siguientes:

- **Accuracy:** es el número de predicciones correctas entre el número total de predicciones.

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

- **Precision:** mide la proporción de las predicciones positivas identificadas correctamente como positivos.

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

- **Recall:** mide cuántos valores positivos fueron correctamente clasificados como positivos.

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

- **F1-Score:** es una métrica que combina la precision y el recall en una sola medida para obtener un valor más objetivo.

$$\text{Precision} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Auc-ROC:** medida de calidad para los problemas de clasificación binaria, representa la tasa de verdaderos positivos (Recall) frente a la tasa de falsos positivos (1 - especificidad).

A estas medidas hay que sumarles las proporcionadas por la organización de la competición *EXIST*¹, para la evaluación de las tareas:

- **Hard-Hard:** Las etiquetas 'hard' se derivan de las etiquetas de los anotadores utilizando umbrales probabilísticos específicos para cada tarea.

- Tarea 4: Se selecciona la clase anotada por más de 3 anotadores.
- Tarea 5: Se selecciona la clase anotada por más de 2 anotadores.

Los elementos sin una clase mayoritaria se eliminan de la evaluación. La métrica oficial es el ICM original, y también se utiliza F1 para comparar.

- **Soft-Soft:** Compara las probabilidades asignadas por el sistema con las asignadas por los anotadores. Como en el caso anterior, ICM-soft se utilizará como la métrica oficial de evaluación.

¹<http://nlp.uned.es/exist2024/>

2.8. Ensemble de Modelos

El ensemble de modelos [19], es una técnica utilizada en el aprendizaje automático para mejorar el rendimiento de los modelos uniendo múltiples modelos predictivos. Algunos de sus beneficios además de la mejora de rendimiento son la reducción del sobreentrenamiento y la versatilidad al poder aplicarse a varios problemas.

Algunas técnicas utilizadas:

- **Bagging**: combina las predicciones mediante votación o promedio.
- **Boosting**: entrena modelos débiles para tratar de mejorar la precisión.
- **Random Forest**: utiliza múltiples árboles de decisión y promedia las predicciones.
- **Stacking**: combina las predicciones de varios modelos utilizando un modelo que aprende a ponderar las contribuciones de cada uno de ellos.

2.9. Aprendizaje con Desacuerdo (*Learning with Disagreement*)

El aprendizaje con desacuerdo (*Learning with Disagreement*) [20], explora como el aprendizaje automático puede beneficiarse de los desacuerdos en los datos etiquetados.

Tradicionalmente, se ha utilizado un enfoque de votación mayoritaria donde uno de los problemas era la posible pérdida de información de algunas anotaciones. Este enfoque hace que se puedan aplicar diferentes estrategias para aprovechar los desacuerdos en los datos etiquetados y así mejorar la precisión de los modelos de clasificación entrenados.

Una de estas técnicas usadas es el ensemble de múltiples modelos entrenados con diferentes subconjuntos de datos. Una posible implementación es aprovechar las características de los anotadores para diferenciar entre estos modelos como vemos en este artículo [21], y así poder mejorar la generalización y calidad del resultado final.

Con esta técnica se pueden tratar las etiquetas de diferentes maneras:

- **Etiquetas Duras (Hard Labels)**: etiquetas binarias que indican la clase de cada instancia.
- **Etiquetas Suaves (Soft Labels)**: etiquetas que representan la probabilidad de que una instancia pertenezca a una clase. Esto permite una representación más flexible de los datos.

2.10. Tecnología y Recursos Utilizados

Para la experimentación del trabajo es necesario el uso de diferentes tecnologías:

-  **Python**: lenguaje de programación utilizado para el desarrollo del código.
-  **Pytorch**: creación y entrenamiento de modelos.
-  **Jupyter Notebook**: desarrollo y ejecución de código.

-  **Google Colab:** desarrollo y ejecución de código.
-  **Hugging Face:** repositorio de los modelos preentrenados.
-  **Nvidia GPU 4070:** entrenamiento de modelos.
-  **Optuna:** optimización de hiperparámetros.
-  **Overleaf:** creación y desarrollo de documentos científicos.
-  **GitHub:** repositorio del proyecto.

3. Metodología, Experimentación y Resultados

En esta sección se describen todos los aspectos relativos a la gestión del proyecto: metodología, preprocesamiento de datos, selección de los modelos, experimentación, evaluación de los resultados y análisis de errores.

3.1. Tarea

Para realizar este proyecto, se ha participado en la 15º edición de CLEF¹, en la competición '*EXIST: sEXism Identification in Social neTworks*'², más concretamente en las tareas 4: '*Sexism Identification in Memes*' y 5: '*Source Intention in Memes*'.

El estudio sobre la desigualdad de género y el sexismo en las redes sociales destaca el uso generalizado de estas plataformas entre los jóvenes, especialmente entre los adolescentes. Se enfoca en cómo se moldean las identidades de género y se construye la igualdad o desigualdad en estas redes. La investigación busca contribuir a la promoción de la igualdad y la erradicación de las actitudes sexistas en las redes sociales.

La capacidad de desarrollar modelos que automaticen la detección de sexismo puede ayudar a diferentes investigaciones a tratar con datos más reales sobre las personas que utilizan este tipo de contenido en redes sociales y a las propias empresas a detectar esta clase de comportamientos en usuarios. Estas tecnologías pueden hacer de las redes sociales un lugar con mayor igualdad.

La organización nos propone investigar sobre el papel que cumplen las redes sociales en la emisión y difusión de mensajes sexistas. Las dos tareas son:

- ***Sexism Identification in Memes***: Es una tarea binaria en la cual se debe clasificar los memes en sexistas y no sexistas.
- ***Source Intention in Memes***: El objetivo de esta tarea es clasificar los memes según la intención del autor. Los memes se clasifican si la intención es directa como 'DIRECT' o si es sarcástica como 'JUDGEMENTAL'.

Las predicciones obtenidas de los modelos para cada tarea son enviados a la organización para establecer un ranking con otros investigadores. Los resultados conseguidos para las tareas fueron:

- **Tarea 4: *Sexism Identification in Memes***: 4º tanto en la medida Hard-Hard como en la Soft-Soft con un valor de 0.5668 y 0.4476 respectivamente.
- **Tarea 5: *Source Intention in Memes***: 2º para la medida Hard-Hard con un ICM-Hard de 0.4119 y 10º para la medida Soft-Soft con un ICM-Soft de 0.2023.

Finalmente, la tarea concluye con el desarrollo de un artículo científico para contribuir al conocimiento en futuras investigaciones.

¹<https://clef2024.imag.fr/>

²<http://nlp.uned.es/exist2024/>

3.2. Descripción de los Datos

La organización proporcionó un único dataset compuesto por una carpeta con los memes y un archivo con las características de cada meme. En las Figuras 3.1 y 3.2, se puede apreciar un ejemplo de cada clase para ambas tareas respectivamente.



Figura 3.1: Ejemplo memes tarea 4



Figura 3.2: Ejemplo memes tarea 5

Las características dadas para cada meme son las siguientes:

- **id_EXIST:** identificador único para el meme.
- **lang:** idioma del meme ('en' o 'es').
- **text:** texto extraído automáticamente del meme.
- **meme:** nombre del archivo que contiene el meme.
- **path_memes:** ruta del archivo que contiene el meme.
- **number_annotation:** número de personas que han anotado el meme.
- **annotators:** identificador único para cada uno de los anotadores.
- **gender_annotation:** género de los diferentes anotadores. Los valores posibles son: 'F' y 'M', para mujer y hombre respectivamente.

- **age_annotation**: grupo de edad de los diferentes anotadores. Los valores posibles son: 18-22, 23-45 y 46+.
- **ethnicity_annotation**: etnicidad autodeclarada de los diferentes anotadores. Los valores posibles son: 'Negro o Afroamericano', 'Hispano o Latino', 'Blanco o Caucásico', 'Multirracial', 'Asiático', 'Indio Asiático' y 'Medio Oriental'.
- **study_level_annotation**: nivel de estudios autodeclarado por los diferentes anotadores. Los valores posibles son: 'Menos que diploma de secundaria', 'Título de secundaria o equivalente', 'Título universitario', 'Máster' y 'Doctorado'.
- **country_annotation** : país autodeclarado donde viven los diferentes anotadores.
- **labels_task4**: conjunto de etiquetas (una para cada uno de los anotadores) que indican si el meme contiene expresiones sexistas o se refiere a comportamientos sexistas o no. Los valores posibles son: 'YES' y 'NO'.
- **labels_task5**: conjunto de etiquetas (una para cada uno de los anotadores) que registran la intención de la persona que creó el meme. Las etiquetas posibles son: 'DIRECT', 'JUDGEMENTAL', '...', y 'UNKNOWN'.
- **split** : subconjunto dentro del conjunto de datos al que pertenece el meme ('TRAIN-MEME', 'TRAIN-MEME' + 'EN'/'ES').

```

"110001": {
  "id_EXIST": "110001",
  "lang": "es",
  "text": "2+2=5 MITO Albert Einstein tenia bajo rendimiento en la escuela. VERDAD 2+2=4 CAN is El feminismo de hoy en dia defiende la estupidez humana y no los derechos de las mujeres quemó ellas afirman ",
  "meme": "110001.jpeg",
  "path_meme": "memes/110001.jpeg",
  "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+", "18-22", "23-45"],
  "ethnicities_annotation": ["Hispanic or Latino", "Hispanic or Latino", "White or Caucasian", "Hispanic or Latino", "Hispanic or Latino"],
  "study_levels_annotation": ["High school degree or equivalent", "Master's degree", "Master's degree", "Bachelor's degree", "Bachelor's degree", "Bachelor's degree"],
  "countries_annotation": ["Mexico", "Spain", "Argentina", "Spain", "Mexico", "Mexico"],
  "labels_task4": ["YES", "YES", "YES", "YES", "YES", "YES"],
  "labels_task5": ["DIRECT", "DIRECT", "DIRECT", "DIRECT", "DIRECT", "DIRECT"],
  "labels_task6": [
    ["IDEOLOGICAL-INEQUALITY", "STEREOTYPING-DOMINANCE", "MISOGYNY-NON-SEXUAL-VIOLENCE"],
    ["IDEOLOGICAL-INEQUALITY"],
    ["IDEOLOGICAL-INEQUALITY"],
    ["IDEOLOGICAL-INEQUALITY"],
    ["IDEOLOGICAL-INEQUALITY"],
    ["IDEOLOGICAL-INEQUALITY"]
  ],
  "split": "TRAIN-MEME_ES"
}

```

Figura 3.3: Ejemplo de características para cada meme

Como se ha dicho anteriormente, se proporcionó un único conjunto de datos por lo que se dividió en un 80% para entrenamiento y un 20% para test. Dentro del conjunto de entrenamiento, a su vez se dividió en un 15% para la validación durante el entrenamiento de los modelos. En la Figura 3.3, se puede ver que para cada meme se obtuvo las diferentes características de los anotadores, así como las etiquetas para cada tarea, siendo seis las valoraciones disponibles para cada una de ellas.

Para tener una aproximación de la cantidad de datos se realizó un sistema de votación mayoritaria. Para la tarea 4, se disponía de un total de 4044 instancias, con 2617 de la clase YES y 1427 de la clase NO. El reparto de entre los conjuntos de datos de entrenamiento, validación y test es el mostrado en la Tabla 3.1.

Tabla 3.1

Distribución de clases para la tarea 4				
Dataset	Total	YES	NO	
Entrenamiento	2749	1810	939	
Validación	486	245	241	
Test	809	476	333	

La tarea 5 dado que solo se necesita valorar entre dos posibles etiquetas ('DIRECT' y 'JUDGEMENTAL'), se realizó el mismo proceso de votación mayoritaria que para la tarea 4 pero los valores de las etiquetas '-' y 'UNKNOWN' quedaron descartados en el proceso. Finalmente, se dispone de un total de 3659 instancias divididas entre 2443 para la clase 'DIRECT' y 1216 de la clase 'JUDGEMENTAL'.

Tabla 3.2

Destribución de clases para la tarea 5				
Dataset	Total	DIRECT	JUDGEMENTAL	
Entrenamiento	2498	1668		830
Validación	440	293		147
Test	721	482		239

3.3. Metodología

La metodología utilizada en este proyecto esta dividida en distintos pasos. Inicialmente, trataremos de dar un enfoque básico utilizado tanto en las imágenes como en los textos para la clasificación de las instancias. Se comprobará cual de ellos tiene mayor eficacia para poder centrar nuestro estudio en la mejor. La forma resultante la estudiaremos desde un enfoque basado en el aprendizaje por desacuerdo (*Learning with Disagreement*), donde se aprovechará las diferentes perspectivas proporcionadas de los anotadores para crear modelos independientes basados en arquitectura *Transformers* para que hagan su predicción sobre los memes de manera individual y unirlas finalmente para encontrar el mejor resultado.

Se estudiará la eficacia de realizar preprocesamiento a los datos y ajuste de los hiperparámetros de los modelos. Una de las novedades introducidas en este trabajo será la utilización de tres datasets, aprovechando que los datos son dados tanto en inglés como en español, se implementarán dos técnicas para su construcción.

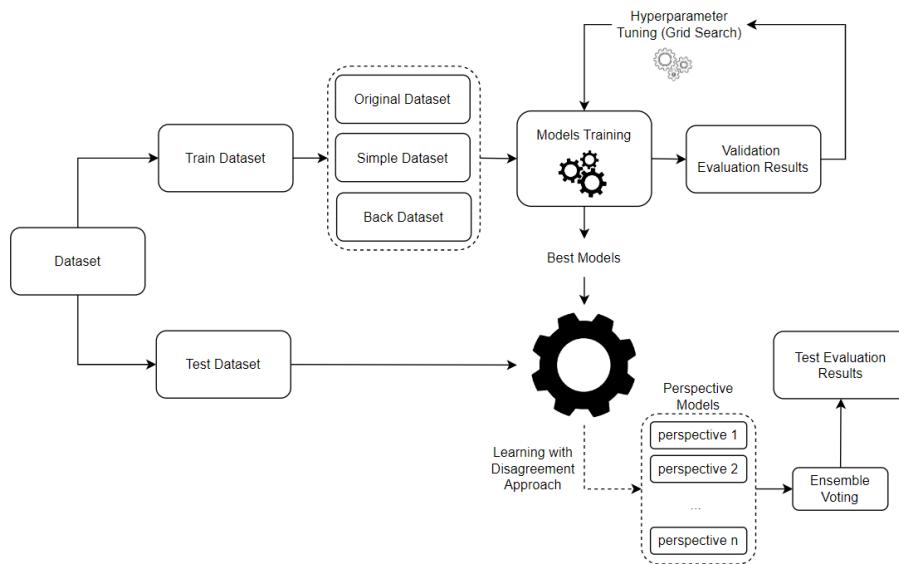


Figura 3.4: Metodología del proyecto

3.4. Estudio de Imágenes y Textos

3.4.1. Selección de Modelos

Como se ha dicho en la sección anterior, inicialmente se estudiarán tanto imágenes como textos por separados, para ver cual de ellos tiene mejor aceptación de los datos y consigue mejores resultados. Para ello, se utilizarán diferentes modelos preentrenados con imágenes y con textos a los que se le realizará un fine-tuning con el dataset proporcionado por la organización.

Los modelos utilizados para el entrenamiento de imágenes son:

- [`beit-base-patch16-224-pt22k-ft22k`](https://huggingface.co/microsoft/beit-base-patch16-224-pt22k-ft22k)³: modelo con arquitectura BEiT, tamaño de parche 16x16 y resolución de imágenes de 224x224 píxeles.
- [`vit-base-patch16-224`](https://huggingface.co/google/vit-base-patch16-224)⁴: modelo con arquitectura ViT, tamaño de parche 16x16 y resolución de imágenes de 224x224 píxeles.

Para el texto los modelos utilizados han de ser multilingüe:

- [`bert-base-multilingual-uncased`](https://huggingface.co/google/bert-base-multilingual-uncased)⁵: variación del modelo BERT (Bidirectional Encoder Representations from Transformers) desarrollado por Google, preentrenado con 104 idiomas.
- [`xlm-roberta-base`](https://huggingface.co/FacebookAI/xlm-roberta-base)⁶: modelo desarrollado por Facebook AI y extensión del modelo RoBERTa (Robustly Optimized BERT Approach), preentrenado con 100 idiomas.

³<https://huggingface.co/microsoft/beit-base-patch16-224-pt22k-ft22k>

⁴<https://huggingface.co/google/vit-base-patch16-224>

⁵<https://huggingface.co/google/bert-base-multilingual-uncased>

⁶<https://huggingface.co/FacebookAI/xlm-roberta-base>

3.4.2. Baseline

Para la comparación de los modelos de imágenes y textos se establecerán unos baselines. Se utilizó la disposición de datos indicadas en la sección 3.2, realizando un undersampling de los datos para conseguir balancear las clases. Los modelos contarán los mismos hiperparámetros por defecto, siendo estos: batch size de 32, learning rate de 3e-5 y weight decay de 0.01. Además, las imágenes fueron redimensionadas a 224x224 píxeles, mientras que para los modelos de textos se contará con un max_length de 128.

Tabla 3.3

Baselines de modelos imágenes para tarea 4

Modelo	F1 Score
BEiT	0.497
ViT	0.534

Tabla 3.4

Baselines de modelos texto para tarea 4

Modelo	F1 Score
BERT	0.6395
XLM-RoBERTa	0.6626

Como se puede apreciar en las Tablas 3.3 y 3.4, para la tarea 4 el estudio de los memes a raíz de los textos es más acertado que en el estudio de las imágenes.

Tabla 3.5

Baselines de modelos imágenes para tarea 5

Modelo	F1 Score
BEiT	0.3543
ViT	0.3807

Tabla 3.6

Baselines de modelos texto para tarea 5	
Modelo	F1 Score
BERT	0.5481
XLM-RoBERTa	0.5520

En las Tablas 3.5 y 3.6, se ve como para la tarea 5 ocurre algo similar a la tarea 4, y es que en ambas el estudio del texto es el más acertado.

3.4.3. Análisis del Estudio

Los baselines tanto de la tarea 4 como de la tarea 5 nos indican que el camino correcto a seguir es el estudio de los datos de texto de manera independiente. Como se puede apreciar en la Figura 3.5, ambos memes pertenecen a la clase sexista de la tarea 4, sin embargo, no tienen nada en común por lo que el clasificador de imágenes no encuentra características similares entre ellas que las relacione.



Figura 3.5: Ejemplo memes sexistas diferentes tarea 4

Por esto, el estudio continuará centrado exclusivamente en el texto de los memes, de donde podremos sacar más información que haga mejores a nuestros clasificadores.

3.5. Preprocesamiento de Datos

3.5.1. Limpieza de Datos

Con el objetivo de eliminar el posible ruido encontrado en el texto de los memes, a todos los modelos se le aplicó por defecto la función *Lower*, que consiste en convertir todas las mayúsculas en minúsculas. Además se probó a aplicar a cada uno de ellos los siguientes métodos:

- Eliminación de enlaces (links).
- Eliminación de usuarios precedidos de arrobas ('@').
- Eliminación de hashtags ('#').

Los métodos fueron aplicados a ambos modelos para cada una de las tareas. Se realizó un undersampling de los datos para que el entrenamiento fuese con los mismos datos que los baselines y así tener una comparación justa. Los resultados obtenidos para el F1-Score se muestran en las Tablas 3.7 y 3.8 para las tareas 4 y 5, respectivamente.

Tabla 3.7

Limpieza de datos para la tarea 4 (F1-Score)				
Modelo	Baseline	Enlaces	Usuarios	Hashtags
BERT	0.6395	0.6031	0.6353	0,6264
XLM-RoBERTa	0.6626	0.6249	0.6583	0.6491

Tabla 3.8

Limpieza de datos para la tarea 5 (F1-Score)				
Modelo	Baseline	Enlaces	Usuarios	Hashtags
BERT	0.5481	0.5330	0.4986	0.5390
XLM-RoBERTa	0.5520	0.5477	0.5519	0.5352

Como se puede apreciar en las Tablas 3.7 y 3.8, ninguno de los modelos de ambas tareas sufren cambios significativos respecto a los conseguidos en el baseline por lo que no se realizará limpieza adicional de los datos.

3.5.2. Tokenización y Codificación

Al trabajar con *Transformers*, antes del entrenamiento cada instancia se debe someter a un proceso de tokenización respecto al modelo seleccionado. Cada modelo tiene su tokenizador y este es importado desde Hugging Face⁷.

```

id_EXIST                                110267
lang                                     es
text          "me autopercibo mujer joven y atractiva
meme                                     110267.jpeg
path_memes                                 memes/110267.jpeg
6
number_annotators
annotators      [Annotator_55, Annotator_56, Annotator_57, Ann...
gender_annotators                         [F, F, F, M, M, M]
age_annotators                            [18-22, 23-45, 46+, 46+, 18-22, 23-45]
ethnicities_annotators                   [Black or African American, White or Caucasian...]
study_levels_annotators                  [Bachelor's degree, Master's degree, Doctorate...]
countries_annotators                     [South Africa, Spain, United Kingdom, Greece, ...]
labels_task4                               NO
labels_task5                               [-, DIRECT, -, -, JUDGEMENTAL, -]
labels_task6                               [[-, [OBJECTIFICATION], [-], [-], [MISOGYNY-N...]
split                                     TRAIN-MEME_ES
Name: 0, dtype: object

```

Figura 3.6: Ejemplo de entrada al proceso de tokenización y codificación

En la Figura 3.6, se muestra como entra una instancia al proceso de tokenización y codificación, mientras que en la Figura 3.7, es la salida de este proceso donde *inputs ids*, es el texto una vez tokenizado.

⁷<https://huggingface.co/>

Figura 3.7: Ejemplo de salida del proceso de tokenización y codificación

3.6. Ajuste de Hiperparámetros

La búsqueda de hiperparámetros [22] es uno de los pasos más importantes para el ajuste del modelo. El parámetro de longitud máxima (*Max size*), el cual determina la longitud máxima de las entradas del modelo fue establecido en 128 tras analizar que en la Figura 3.8 es el mejor valor para este parámetro.

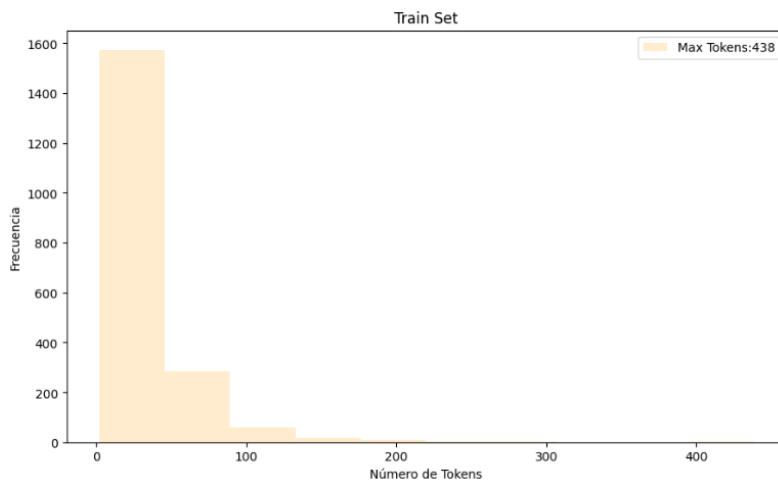


Figura 3.8: *Max Length* para tokenización

Se evaluaron diferentes combinaciones de los siguientes hiperparámetros:

- **Tamaño de lote (Batch size)**: número de instancias que se procesan en cada iteración del entrenamiento en paralelo.
 - **Tasa de aprendizaje (Learning rate)**: regulador de magnitud de los cambios de los parámetros durante el entrenamiento.
 - **Decaimiento de peso (Weight decay)**: regularizador del sobreajuste de los parametros del modelo durante el entrenamiento.

Para la búsqueda de los mejores hiperparámetros de cada modelo se realizó una experimentación de diferentes combinaciones, donde se ajustó el número de datos 1000 instancias por cada clase, para reducir el coste temporal. Esta fue realizada con la librería de Python Optuna [23], la cual nos permite hacerlo de manera eficiente y flexible. El espacio de hiperparámetros utilizado fue el siguiente.

Tabla 3.9

Espacio de hiperparámetros	
Hiperparámetros	Valores
Batch Size	[8, 16, 32]
Learning Rate	[1e-05, 3e-05, 5e-05]
Weight Decay	[0.01, 0.1]

En las Tablas 3.10 y 3.11, se puede ver los mejores hiperparámetros para cada modelo en las dos tareas.

Tabla 3.10

Mejores hiperparámetros para la tarea 4		
Hiperparámetro	BERT	XLM-RoBERTa
Batch Size	16	16
Learning Rate	3e-05	1e-05
Weight Decay	0.1	0.01

Tabla 3.11

Mejores hiperparámetros para la tarea 5		
Hiperparámetro	BERT	XLM-RoBERTa
Batch Size	32	32
Learning Rate	3e-05	1e-05
Weight Decay	0.1	0.01

3.7. Enfoque Aprendizaje con Desacuerdo

El dataset proporcionado por la competición contiene muchas características de los diferentes anotadores, lo que provoca que se tenga mucha información para ser utilizada de diferentes formas. El enfoque basado en perspectivas hace que se pueda recopilar información de forma más específica pero a su vez se cuenta con un menor número de datos para cada una de ellas y es computacionalmente más costoso. Aún así se explorará el estudio de distintas perspectivas de manera individual.

Las perspectivas por cada característica del dataset son:

- **Género:** 'F' y 'M'.
- **Edad:** '18-22', '23-45', '46+'.
- **Étnias:** 'Black or African America', 'Hispano or Latino', 'White or Caucasian', 'Multiracial', 'Asian', 'Asian Indian' and 'Middle Eastern'.
- **Nivel de estudios:** 'Less than high school diploma', 'High school degree or equivalent', 'Bachelor's degree', 'Master's degree' and 'Doctorate'.
- **País:** diferentes países los anotadores.

Se muestran para cada tarea las perspectivas que tienen datos suficientes en el total del dataset dado por la organización, sin la separación realizada previamente. En la tabla 3.12 se puede ver las más interesantes para la tarea 4.

Tabla 3.12

Balanceo de datos de las perspectivas de la tarea 4

Perspectiva	YES	NO	Total
F	2448	1596	4044
M	2174	1870	4044
18-22	2728	1316	4044
23-45	2728	1316	4044
46+	3147	897	4044
Bachelor's	2619	1425	4044
Master	3112	932	4044
High school	2692	1352	4044
White	2575	1469	4044
Hispano	3054	990	4044
Mexico	3199	845	4044
Argentina	3991	53	4044
Spain	3338	706	4044

La Tabla 3.13, nos muestra lo mismo para la tarea 5.

Tabla 3.13

Balanceo de datos de las perspectivas de la tarea 5

Perspectiva	DIRECT	JUDGEMENTAL	Total
F	2160	1200	3360
M	2055	1148	3203
18-22	1749	968	2717
46+	2011	1120	3131
23-45	1864	1083	2947
Bachelor's	1942	1105	3047
Master	1104	549	1653
High school	1470	872	2342
White	2217	1101	3318
Hispano	1290	582	1872
Mexico	877	444	1321
Argentina	61	21	82
Spain	1039	345	1384

En ambas tablas, se ha resaltado las ocho perspectivas elegidas, siendo estas las que computacionalmente nos ha permitido entrenar las herramientas de las que se dispuso para la experimentación. Al entrenar estas perspectivas se ha utilizado la técnica de undersampling para balancear los datos, por lo tanto las elecciones vienen dadas por el mayor número de datos de la clase minoritaria para cada una de ellas.

3.8. Construcción de los Datasets

Como se puede ver en la sección anterior, para un trabajo de clasificación se dispone de un número muy limitados de datos, por ello, se buscaron distintas técnicas para poder ampliar la cantidad de estos. Aprovechando que los textos están en dos idiomas (inglés y español), se tradujo cada instancia al idioma opuesto, realizando así una traducción simple [24] y formando un nuevo conjunto de datos al unirlos al dataset original dado por la organización.

Otra técnica empleada fue la retrotraducción [25], donde cada instancia fue traducida a otro idioma (en este caso, alemán) y vuelta a traducir a su idioma original. En este caso, para la traducción de todas las instancias se utilizó CHAT GPT [26], dada su eficacia para este trabajo. Finalmente, se unió al dataset original y así se formó otro de los conjuntos de datos.

Los tres conjuntos de datos utilizados durante la experimentación serán:

- **Original:** Dataset original proporcionado por la organización.
- **Simple:** Dataset original más extensión de traducción simple.
- **Back:** Dataset original más extensión de retrotraducción.

3.9. Entrenamiento de los Modelos

El entrenamiento de los modelos fue realizado de forma individual, es decir, cada perspectiva fue entrenada por separado. Para ello se utilizó el objeto Trainer⁸ de Hugging Face⁹, ya que es una herramienta muy buena para el entrenamiento de *Transformers*, optimizado para facilitar el proceso de entrenamiento, donde se incluyen numerosas características.

Cada perspectiva se entrenó con los tres datasets indicados en la sección 3.8, Construcción de Datasets, usando la técnica de undersampling para balancear los datos. Los modelos utilizados fueron: BERT¹⁰ y XLM-RoBERTa¹¹. Para cada modelo se usarón los mejores hiperparámetros vistos previamente en la sección 3.6, Ajuste de Hiperparámetros.

En la configuración del entrenamiento se optó por el optimizador Adam [27] ya que permite adaptar las tasas de entrenamiento dinámicamente, combinando los beneficios del descenso por gradiente y RMS-Drop, lo que hace una convergencia rápida y robusta ante el escalado. El número de épocas seleccionado fue de 15, mientras que para la monitorización del entrenamiento, se usó la función *Callback*, para controlar el flujo del entrenamiento, con una paciencia de 3 épocas.

La métrica utilizada para la parada fue el F1-Score como podemos ver en la Figura 3.9.

[580/870 02:18 < 01:09, 4.16 it/s, Epoch 10/15]												
Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall	Auc	F1 Minoritaria	F1 Mayoritaria	Prec	Rec	
1	No log	0.687412	0.550459	0.355030	0.275229	0.500000	0.500000	0.710059	0.000000	0.550459		
2	No log	0.687956	0.516820	0.468063	0.488407	0.491270	0.491270	0.629108	0.307018	0.546183		
3	No log	0.736616	0.519878	0.504282	0.507557	0.507143	0.507143	0.592208	0.416357	0.554030		
4	No log	0.748550	0.529052	0.526283	0.526500	0.526701	0.526701	0.562500	0.490066	0.564276		
5	No log	0.978606	0.550459	0.538796	0.541457	0.539909	0.539909	0.612137	0.465455	0.571375		
6	No log	1.224952	0.541284	0.540937	0.543691	0.544048	0.544048	0.553571	0.528302	0.574068		
7	No log	1.370306	0.568807	0.567188	0.567758	0.568424	0.568424	0.593660	0.540717	0.588401		
8	No log	1.866820	0.562691	0.562282	0.564800	0.565363	0.565363	0.575668	0.548896	0.586767		
9	0.412100	2.056860	0.553517	0.550452	0.550485	0.550794	0.550794	0.587571	0.513333	0.577754		
10	0.412100	2.351678	0.547401	0.546174	0.547247	0.547732	0.547732	0.569767	0.522581	0.576103		

Figura 3.9: Ejemplo visualización durante el entrenamiento

⁸https://huggingface.co/docs/transformers/main_classes/trainer

⁹<https://huggingface.co/>

¹⁰<https://huggingface.co/google-bert/bert-base-multilingual-uncased>

¹¹<https://huggingface.co/FacebookAI/xlm-roberta-base>

3.10. Validación y Evaluación de los Modelos

3.10.1. Evaluación de Perspectivas

El entrenamiento realizado de manera individual para cada perspectiva produjo los siguientes resultados para la métrica F1-Score:

Tabla 3.14

Resultados F1-Score para cada perspectiva de la tarea 4

Modelo	Dataset	M	F	23-45	18-22	46+	Bachelor's	High school	White
BERT	Original	0.6580	0.6205	0.5920	0.6360	0.6231	0.6131	0.6270	0.6611
BERT	Simple	0.6213	0.6191	0.6255	0.6417	0.6223	0.6170	0.6150	0.6679
BERT	Back	0.6613	0.5699	0.6710	0.6507	0.5967	0.6349	0.6228	0.6393
XLM-RoBERTa	Original	0.6251	0.6264	0.6700	0.6493	0.6270	0.6559	0.6553	0.6471
XLM-RoBERTa	Simple	0.6445	0.6457	0.6744	0.6469	0.6425	0.6375	0.6387	0.6768
XLM-RoBERTa	Back	0.6468	0.6305	0.6637	0.6398	0.6299	0.6431	0.6501	0.6688

Tabla 3.15

Resultados F1-Score para cada perspectiva de la tarea 5

Modelo	Dataset	M	F	23-45	18-22	46+	Bachelor's	High school	White
BERT	Original	0.5375	0.5805	0.5609	0.5132	0.5500	0.5357	0.5402	0.5616
BERT	Simple	0.5148	0.5170	0.5324	0.4991	0.4923	0.5033	0.5222	0.5204
BERT	Back	0.5225	0.5159	0.5314	0.5460	0.5649	0.5516	0.5095	0.5567
XLM-RoBERTa	Original	0.5322	0.5275	0.5474	0.5311	0.4906	0.5317	0.5372	0.5738
XLM-RoBERTa	Simple	0.5472	0.4938	0.5572	0.5517	0.5589	0.5502	0.5390	0.5635
XLM-RoBERTa	Back	0.5299	0.5336	0.5449	0.5591	0.5760	0.5432	0.5479	0.5072

Como se puede ver en las Tablas 3.14 y 3.15, se resaltan los modelos seleccionados para cada perspectiva. Dado que se estudia los modelos por separado, se elige el mejor modelo para cada perspectiva basado en el conjunto de datos utilizado para su entrenamiento. El Modelo 1 para la tarea 4 está compuesto por la perspectiva 'M' con el conjunto de datos de entrenamiento 'Back', 'F' con 'Original', '23-45' con 'Back', '18-22' con 'Back', '46+' con 'Original', 'Bachelor's' con 'Back', 'High school' con 'Original' y 'White' con 'Simple'.

Se hicieron modelos para cada tarea, siguiendo las siguientes arquitecturas.

- **Modelo 1 and Modelo 4:** Modelos BERT más eficientes de cada perspectiva para la Tarea 4 y la Tarea 5 respectivamente.
- **Modelo 2 and Modelo 5:** Modelos XLM-RoBERTa más eficientes de cada perspectiva para la Tarea 4 y la Tarea 5 respectivamente.
- **Modelo 3 and Modelo 6:** Combinación de modelos BERT/XLM-RoBERTa más eficientes de cada perspectiva para la Tarea 4 y la Tarea 5 respectivamente.

3.10.2. Ensemble de Modelos

En esta sección, se describe como se unen los modelos para hacer una predicción conjunta a partir de sus predicciones individuales. Para la optimización de los resultados, se realizó un sistema de pesos donde

se le asigna cada uno de ellos a las predicciones decada perspectiva. Fue realizado mediante un proceso de búsqueda comparando el F1 conjunto. Con este sistema lo que se busca es que todas las predicciones individuales aporten a la predicción final, pero las mejores lo hagan en mayor medida.

Tabla 3.16

Espacio de pesos para cada predicción individual	
Pesos	
{0.5, 0.75, 1, 1.25, 1.5, 1.75}	

En la Tabla 3.16, se pueden ver los valores posibles de los pesos de cada perspectiva, teniendo en cuenta que la combinación de todos tienen que sumar ocho. Se realizó una búsqueda exhaustiva de 98812 combinaciones posibles.

Las mejores combinaciones de los modelos para cada tarea fueron:

Tabla 3.17

Mejores combinaciones para la tarea 4

Número	Modelo	M	F	23-45	18-22	46+	Bachelor's	High school	White	Overall F1 score
1	Modelo 1	1.25	0.75	1.25	0.75	0.5	0.5	1.5	1.5	0.7294
2	Modelo 1	1.25	0.75	1.25	0.75	0.5	0.5	1.25	1.75	0.7288
3	Modelo 2	1.75	0.5	1.5	0.75	0.75	0.5	0.5	1.75	0.7052
4	Modelo 2	1.75	0.5	1.25	0.75	0.75	0.5	0.75	1.75	0.7029
5	Modelo 3	0.75	0.5	1.75	1.75	0.5	1	1	0.75	0.7224
6	Modelo 3	1.5	1	0.75	1.75	0.5	0.5	1.5	0.5	0.7187

Tabla 3.18

Mejores combinaciones para la tarea 5

Número	Modelo	M	F	23-45	18-22	46+	Bachelor's	High school	White	Overall F1 score
7	Modelo 4	1.25	0.5	1.75	0.5	1	1	1.5	0.5	0.6352
8	Modelo 4	1	0.5	1.75	0.5	1	1	1.75	0.5	0.6329
9	Modelo 5	0.5	1	1.75	0.75	1.75	0.75	0.5	1	0.6061
10	Modelo 5	0.5	1	1.5	1	1.5	0.75	0.5	1.25	0.6049
11	Modelo 6	1.5	1.5	1.5	0.75	1.25	0.5	0.5	0.5	0.6147
12	Modelo 6	1.75	1.25	1.5	0.5	1	0.5	1	0.5	0.6114

En las Tablas 3.17 and 3.18, se resaltan los tres mejores modelos para cada tarea ya que este fue el número de envíos disponibles para la competición. Los modelos enviados para la tarea 4 son:

- **Run 1 (I2C-Huelva_1)**: Modelo 1 con balance de pesos número 1.
- **Run 2 (I2C-Huelva_2)**: Modelo 1 con balance de pesos número 2.
- **Run 3 (I2C-Huelva_3)**: Modelo 3 con balance de pesos número 5.

Para la tarea 5, se debía escoger los resultados de una de las Runs de la tarea 4 y volver a clasificarlo con los modelos de la tarea 5. Las combinaciones elegidas fueron:

- **Run 4 (I2C-Huelva_1)**: Run 1 con Modelo 4 y balance de pesos número 7.
- **Run 5 (I2C-Huelva_2)**: Run 3 con Modelo 4 y balance de pesos número 7.
- **Run 6 (I2C-Huelva_3)**: Run 2 con Modelo 4 y balance de pesos número 8.

3.11. Análisis de Errores

En esta sección, se examinarán los errores de los modelos a través del análisis de sus matrices de confusión. Este enfoque permitirá una comprensión detallada del rendimiento de los modelos, identificando tanto sus aciertos como sus fallos al clasificar las muestras. Esta evaluación crítica proporcionará información valiosa para mejorar la precisión y la fiabilidad de los modelos, contribuyendo así al avance del estudio.

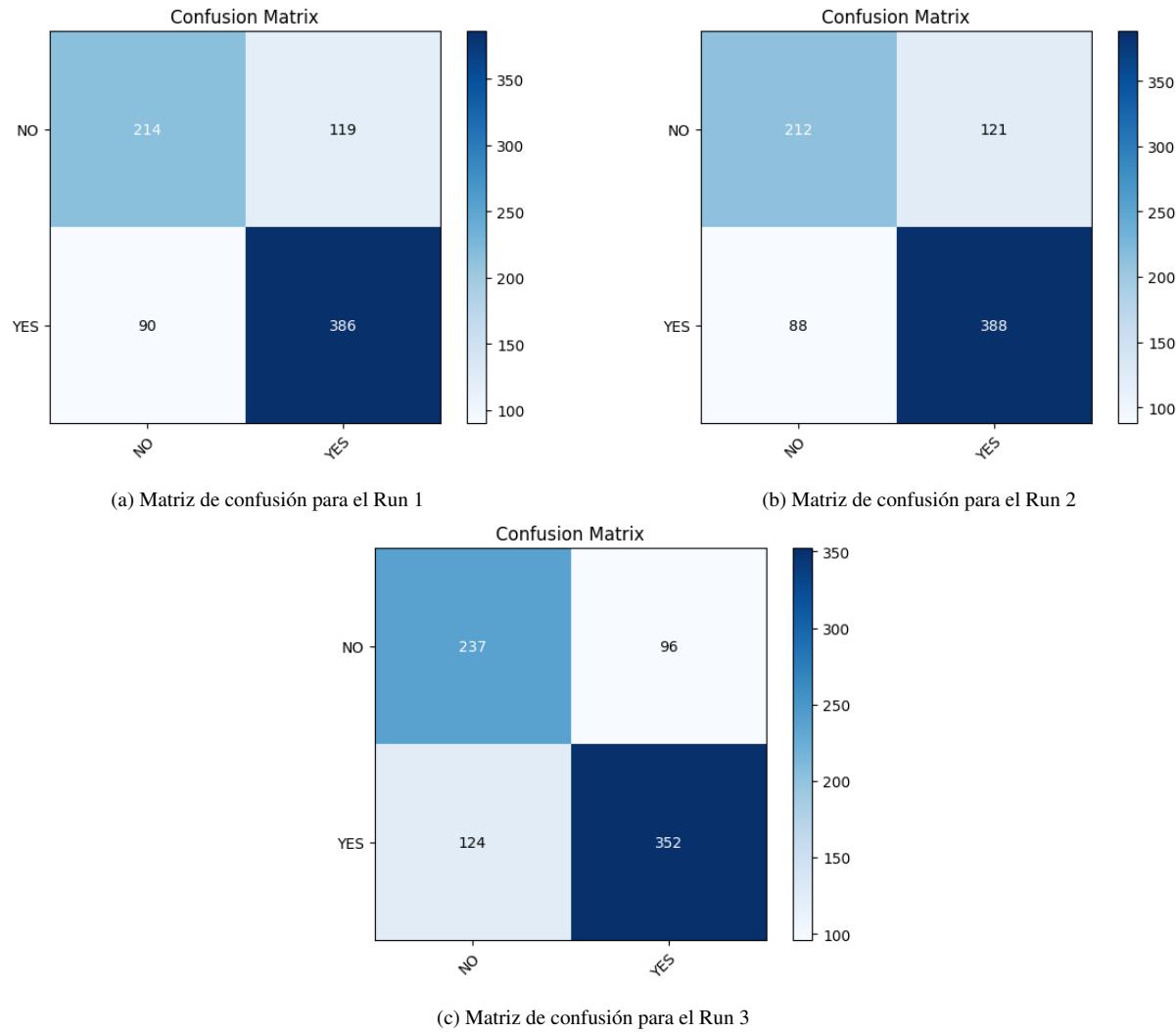


Figura 3.10: Matrices de confusión para la tarea 4

Para la tarea 4, todas las figuras muestran patrones de rendimiento similares, con una proporción de predicciones correctas (TP y TN) alta en comparación con las incorrectas (FP y FN). Esto sugiere una capacidad consistente de los modelos para clasificar con precisión muestras de clases positivas y negativas. Sin embargo, las diferencias entre los modelos revelan tendencias distintas. La Figura 3.10c muestra un mayor número de verdaderos positivos (TP) en comparación con las Figuras 3.10a y 3.10b, lo que indica una capacidad potencialmente mejor del modelo mixto BERT/XLM-RoBERTa para identificar instancias de la clase positiva.

Las Figuras 3.10a and 3.10b demuestran tendencias similares en falsos positivos (FP) y falsos negativos (FN), mientras que la Figura 3.10c, muestra una proporción ligeramente mayor de falsos negativos (FN). Estas discrepancias podrían deberse a las variaciones en las arquitecturas de los modelos (utilizar únicamente BERT frente al modelo mixto BERT/XLM-RoBERTa) y a las características específicas del

conjunto de datos y los procesos de entrenamiento. Combinando estas observaciones, se puede informar la investigación futura sobre la selección y optimización de modelos para tareas de clasificación.

Tabla 3.19

Ejemplos de etiquetado tarea 4

Texto	Test	Run 1	Run 2	Run 3
metro q estilo de vida alexa recomienda a una madre asesinar a sus hijos: amazon pide disculpas por .eror en la configuración.el asistente inteligente ofreció una respuesta polémica cuando una mujer le preguntó sobre cómo evitar que los niños rían.alex le pusieron ese nombre por no llamarla skynet más en cuantara-zon.com	0	1	1	1
ME DIJO QUE ME FUERA A FREGAR memegenerator.es	1	0	0	0

La Tabla 3.19 muestra la dificultad de clasificar con precisión ciertos textos. En el primer ejemplo, el texto aborda temas controvertidos y entidades específicas (p. ej., Amazon, Alexa), lo que puede llevar a una clasificación errónea debido a la falta de contexto. En el segundo ejemplo, se demuestra la variedad de temas y la presencia de elementos humorísticos que pueden complicar la tarea de clasificación automatizada.

Para la Tarea 5, las Runs 4 y 5 son idénticas, mientras que la Ejecución 6 se basa en el mismo modelo pero con diferentes pesos utilizados para la predicción. Por lo tanto, se comparará las matrices de confusión de las Runs 4 y 6.

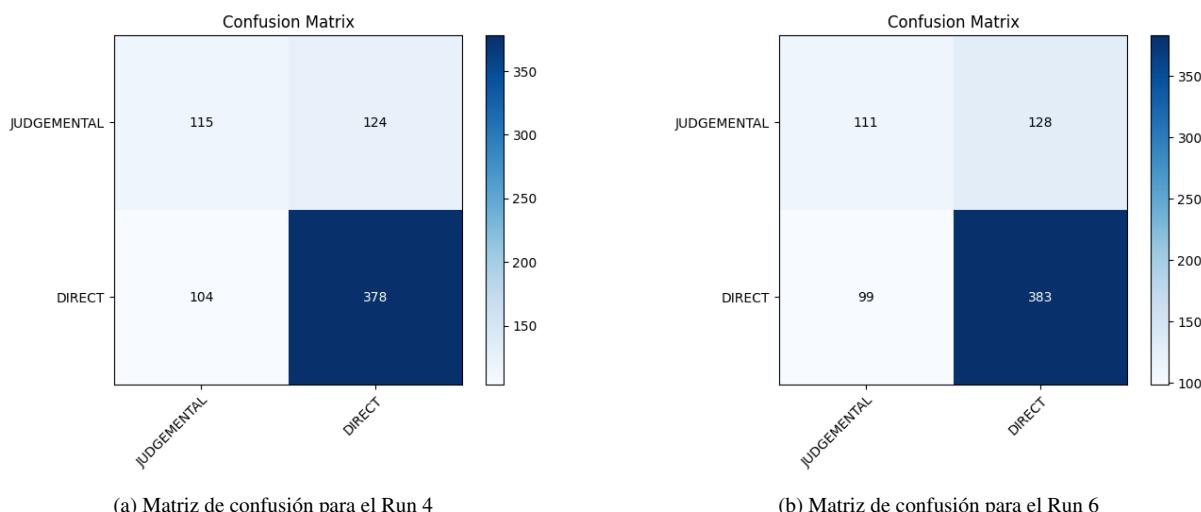


Figura 3.11: Matrices de confusión para la tarea 5

Ambas Figuras 3.11a y 3.11b, se basan en la misma arquitectura BERT. Las diferencias en la distribución de errores y el sistema de valores del modelo final sugieren que los modelos se han entrenado de manera ligeramente diferente. Sin embargo, comparten similitudes fundamentales debido a su base común en BERT y su estructura de matriz idéntica.

Tabla 3.20

Ejemplos de etiquetado para la tarea 5

Texto	Test	Run 4	Run 6
La mecánica es solo para hombres, toma mi bolso, sé más que tú de motores, putito.	0	1	1
yet men don't have the same rights as women like the right to share their opinion on abortion	1	0	0

En el primer ejemplo de la Tabla 3.20, la clasificación errónea de este texto podría atribuirse a la falta de consideración del contexto cultural, social y lingüístico, así como a la incapacidad de un algoritmo automatizado para captar matices en el tono y la intención comunicativa. En el segundo ejemplo, el texto discute los derechos de género con un enfoque específico en la disparidad de opiniones sobre el aborto, lo que introduce temas sensibles y dependientes del contexto. Estos casos demuestran el desafío de clasificar textos con un vocabulario similar pero con contextos diferentes o contenido fragmentado.

3.12. Resultados Oficiales de la Competición

Esta sección presenta los resultados obtenidos en la competición, detallando el rendimiento de los mejores modelos para ambas tareas. Como ya se ha dicho antes cada tarea tiene dos formas de evaluación, Hard-Hard y Soft-Soft, por lo que hay dos clasificaciones para cada una de ellas.

Tal y como se han planteado los modelos dan un porcentaje de la clase principal, en el caso de la tarea 4 sería la clase 'YES' y para la tarea 5, la clase 'DIRECT'. Este porcentaje fue el enviado para la medida Soft-Soft, mientras que para la medida Hard-Hard, en caso de que el porcentaje superase o fuese igual al 50% pertenecería a la clase principal de cada tarea. En las Tablas 3.21 y 3.22 observamos los rankings para la tarea 4, pudiendo decir que el enfoque utilizando las perspectivas de los anotadores fue bastante bueno.

Tabla 3.21

Ranking de participantes para la tarea 4 Hard-Hard

Rank	Run	ICM-Hard	ICM-Hard Norm	F1_YES
1	RoJiNG-CL_3	0.3182	0.6618	0.7642
2	RoJiNG-CL_2	0.2272	0.6155	0.7437
3	RoJiNG-CL_1	0.1863	0.5947	0.7274
4	I2C-Huelva_2	0.1313	0.5668	0.7241
5	I2C-Huelva_1	0.1166	0.5593	0.7154
-	-	-	-	-
9	I2C-Huelva_3	0.0987	0.5502	0.6933
-	-	-	-	-
53	melialo-vcassan_1	-0.8109	0.0876	0.5316

Tabla 3.22

Ranking de participantes para la tarea Soft-Soft				
Rank	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
1	Victor-UNED_1	-0.2925	0.4530	1.1028
2	Victor-UNED_2	-0.3135	0.4496	1.2834
3	Elias&Sergio_1	-0.3225	0.4482	0.9903
4	I2C-Huelva_3	-0.3263	0.4476	1.5189
5	I2C-Huelva_1	-0.3390	0.4455	1.4096
6	I2C-Huelva_2	-0.3446	0.4446	1.4112
-	-	-	-	-
37	CNLP-NITS-PP_1	-2.6987	0.0662	1.3445

Para la tarea 5, el ranking conseguido para la medida Hard-Hard fue muy bueno, quedando en 2º posición, mientras que en la medida Soft-Soft no se terminó de alcanzar una buena posición, quedando 10º de 17 competidores.

Tabla 3.23

Ranking de participantes para la tarea 5 Hard-Hard				
Rank	Run	ICM-Hard	ICM-Hard Norm	Macro F1
1	Victor-UNED_1	-0.2397	0.4167	0.3873
2	I2C-Huelva_2	-0.2535	0.4119	0.4761
3	Victor-UNED_2	-0.2668	0.4073	0.3850
4	I2C-Huelva_3	-0.2772	0.4036	0.4714
5	I2C-Huelva_1	-0.2880	0.3999	0.4714
-	-	-	-	-
22	epistemologos_1	-8.7012	0.0000	0.0557

Tabla 3.24

Ranking de participantes para la tarea 5 Soft-Soft				
Rank	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
1	Victor-UNED_2	-1.2453	0.3676	1.6235
-	-	-	-	-
8	melialo-vcassan_3	-2.0653	0.2804	1.5295
9	melialo-vcassan_1	-2.6821	0.2148	1.6291
10	I2C-Huelva_3	-2.7996	0.2023	3.9604
11	I2C-Huelva_2	-2.7997	0.2023	3.9857
12	I2C-Huelva_1	-2.8007	0.2022	3.9735
-	-	-	-	-
17	Penta-ML_2	-5.9832	0.0000	5.4845

4. Conclusiones y Trabajo Futuro

En esta sección se exponen las conclusiones obtenidas una vez realizado el estudio, así como posibles consideraciones a tener en cuenta para trabajos futuros. Finalmente, se muestra la planificación temporal seguida durante el proyecto.

4.1. Conclusiones

En este estudio se exploró la identificación de sexismo e intención del autor en memes para la competición *EXIST 2024*. Se aplicaron técnicas como *Deep Learning*, *Transformers* y procesamiento del lenguaje natural para la experimentación de las tareas, donde mostraron cumplir una función importante durante la etapa de experimentación.

El proyecto se enfocó principalmente en comparar diferentes enfoques a la hora de trabajar con los datos. Se demostró como el uso de las imágenes no termina de ser un enfoque correcto ya que las imágenes no tienen un contenido característico en ellas que pueda hacer que los modelos consigan un buen funcionamiento. Respecto al estudio del texto, se utilizó un enfoque basado en aprendizaje con desacuerdo (*Learning with Disagreement*) el cual permitió entrenar diferentes perspectivas para después unirlas mediante ensemble de modelos.

Algunas de las limitaciones encontradas durante el estudio fue el número de datos disponibles que proporcionó la organización, ya que era muy reducido para que lo suele implicar normalmente los modelos de clasificación. Una de las medidas aplicadas fue la construcción de dos datasets con los que entrenar los modelos, ya que permitió tener un mayor abanico de posibilidades durante la experimentación.

Los resultados obtenidos durante el entrenamiento fueron bastante positivos, el mejor modelo final consiguió 0.068 y 0.083 de F1-Score más que el baseline establecido usando la votación mayoritaria con los hiperparámetros de los modelos optimizados. Respecto a la competición, fueron bastante positivos a excepción de la medida Soft-Soft para la tarea 5 donde no se consiguió un buen ranking. Estos resultados reafirman observar que se ha utilizado un buen enfoque para la detección de sexismo en redes sociales para la lucha por la igualdad, pero también la dificultad que siguen teniendo los modelos de clasificación para detectar sarcasmos o ironías en memes.

El desarrollo de un paper científico para la competición permitió aprender a desarrollar documentos en Latex.

4.2. Trabajo Futuro

Tras la investigación se han visto posibles áreas de mejora para futuras investigaciones. El problema principal detectado durante el estudio ha sido la escasez de datos ya que se descartó el uso de imágenes durante la investigación.

Una de las propuestas para futuros trabajos es estudiar de manera más exhaustiva las imágenes, ya que, aunque no sean tan relevantes a la hora de obtener información como se ha demostrado durante este estudio, sí que podría incluir información interesante. Se podrían entrenar modelos para analizar las instancias que el clasificador etiqueta correctamente buscando características que solo se encuentran en

las imágenes. Este clasificador podría ser incluido como parte del ensemble de modelos realizado en el trabajo asignándole un peso adecuado.

El balanceo de datos de cada perspectiva de manera individual, podría ser estudiado exhaustivamente por separado y aplicar diferentes técnicas para su balanceo para su equilibrado.

4.3. Planificación Temporal del Trabajo Realizado

En este apartado se muestra el tiempo empleado en el proyecto, así como el tiempo empleado en las secciones más importantes, como se muestra en la Tabla 4.1.

Tabla 4.1

Planificación Temporal del Trabajo	
Tarea	Horas
Estudio de la tarea EXIST 2024	5
Aprendizaje de los conceptos necesarios para abordar el TFG - Conceptos sobre el aprendizaje automático - Redes neuronales - Transformers - Transfer Learning - Fine Tuning - Procesamiento del Lenguaje Natural - Aprendizaje con Desacuerdo - Métricas de evaluación de modelos	50
Aprendizaje de la tecnología necesaria - Librería Pytorch - Modelos de transfer learning para clasificación NLP - Librería NLTK - Modelos multimodales - Librería Optuna - Latex	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	120
Elaboración de paper y correcciones	40
Elaboración de la memoria del proyecto	45
Total	340

Bibliografía

- [1] Zhengkai Tu, Thijs Stuyver, and Connor W Coley. Predictive chemistry: machine learning for reaction deployment, reaction development, and reaction discovery. *Chemical science*, 14(2):226–244, 2023.
- [2] Esperanza Manrique Rojas. Machine learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E28):586–599, 2020.
- [3] Francisco Rodríguez-Sánchez, Jorge Carrillo-de Albornoz, and Laura Plaza. Automatic classification of sexism in social networks: An empirical study on twitter data. *IEEE Access*, 8:219563–219576, 2020.
- [4] JE Sierra-García and M Santos. Redes neuronales y aprendizaje por refuerzo en el control de turbinas eólicas. *Revista Iberoamericana de Automática e Informática industrial*, 18(4):327–335, 2021.
- [5] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.
- [6] Carmelo Bonilla Carrión. Redes convolucionales. 2020.
- [7] Isis Bonet Cruz, Sain Salazar Martínez, Abdel Rodríguez Abed, Ricardo Grau Ábalos, and María Matilde García Lorenzo. Redes neuronales recurrentes para el análisis de secuencias. *Revista Cubana de Ciencias Informáticas*, 1(4):48–57, 2007.
- [8] Special issue: Artificial intelligence and computer vision applications. *Computers*. Accessed: 2024-06-15.
- [9] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 111–118, 2010.
- [10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [11] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [14] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [16] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.
- [17] Xiangnan Yin, Weihai Chen, Xingming Wu, and Haosong Yue. Fine-tuning and visualization of convolutional neural networks. In *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 1310–1315. IEEE, 2017.
- [18] DataBit AI. Métricas de evaluación en machine learning. <https://databitai.com/machine-learning/metricas-de-evaluacion-en-machine-learning/>, consultado en 2024.
- [19] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 8(4):e1249, 2018.
- [20] Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470, 2021.
- [21] Simona Frenda, Alessandro Pedrani, Valerio Basile, Soda Marem Lo, Alessandra Teresa Cignarella, Raffaella Panizzon, Cristina Sánchez-Marco, Bianca Scarlini, Viviana Patti, Cristina Bosco, et al. Epic: Multi-perspective annotation of a corpus of irony. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13844–13857, 2023.
- [22] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.
- [23] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.
- [24] Yu Li, Xiao Li, Yating Yang, and Rui Dong. A diverse data augmentation strategy for low-resource neural machine translation. *Information*, 11(5):255, 2020.
- [25] Djamila Romaissa Beddiar, Md Saroor Jahan, and Mourad Oussalah. Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24:100153, 2021.
- [26] Yuan Gao, Ruili Wang, and Feng Hou. How to design translation prompts for chatgpt: An empirical study. *arXiv e-prints*, pages arXiv–2304, 2023.
- [27] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anexos

Anexo A.- Repositorios de Código en GitHub

Uno de los aspectos destacados de este trabajo ha sido la creación de un repositorio de códigos en *GitHub*, situados en el siguiente enlace: <https://github.com/alvarxcarrillo/Deteccion-De-Sexismo-En-Memes-Con-Transformers-Y-Learning-With-Disagreement>. Su objetivo es proporcionar a otros investigadores el desarrollo de este proyecto para que pueda ser utilizado como punto de estudio a incorporar en sus investigaciones. A continuación, se aporta una resumen del contenido basado en el archivo *README.md*.

Descripción del Repositorio

Este repositorio contiene el código fuente y los recursos necesarios para aplicar la experimentación del Trabajo de Fin de Grado 'Detección de Sexismo e Intención del Autor en Memes Basado en Transformers Utilizando un Enfoque de Aprendizaje con Desacuerdo'. Centrado en usar técnicas de aprendizaje automático, procesamiento del lenguaje natural y *Transformers* para la detección de sexismo e intención del autor en memes utilizando un enfoque de aprendizaje con desacuerdo (*Learning with Disagreement*).

Esta centrada en el desarrollo y experimentación de las tareas 4 y 5 de la competición '*EXIST 2024: sEXism Identification in Social netWorks*'.

Contenido del repositorio

En el repositorio se puede encontrar el siguiente contenido:

- Descripción de tareas: Incluye el PDF dado por la organización donde se explica todo el contenido relacionado con las tareas.
- Datasets: Carpeta donde se almacenan los diferentes dataset utilizados durante la experimentación y el test no etiquetado dado por la organización.
- Preprocesamiento de datos: Carpeta donde se encuentra los cuadernos para la optimización de hiperparámetros y la construcción de nuevos conjuntos de datos.
- Notebooks de Entrenamiento y Evaluación: Contiene los cuadernos desarrollados en *Jupyter Notebook* con el proceso de entrenamiento utilizado para el enfoque básico de votación mayoritaria tanto de texto como de imágenes. También están los cuadernos usando el enfoque de aprendizaje con desacuerdo basado en el estudio de las perspectivas de los anotadores.
- Modelos: Incluye carpetas con los modelos finales para la evaluación de las tareas 4 y 5.
- Evaluador: Carpeta que contiene los cuadernos desarrollados para la evaluación del dataset no etiquetado de ambas tareas.
- Resultados: Archivo csv donde se almacen los diferentes resultados obtenidos durante la experimentación.

- Documentación: Incluye archivos como la memoria y el artículo científico, lo que proporciona toda la información sobre las fases de desarrollo de la investigación.

Uso del Repositorio

El acceso y uso del repositorio se ofrecen bajo una licencia completamente abierta, permitiendo a los investigadores utilizar los recursos contenidos de manera libre y sin restricciones para sus propios estudios y trabajos de investigación.

Anexo B.- Artículo Científico

I2C-UHU at EXIST 2024: Transformer-Based Detection of Sexism and Source Intention in Memes Using a Learning with Disagreement Approach.

Workings Notes EXIST 2024: sEXism Identification in Social neTworks.

I2C-UHU at EXIST 2024: Transformer-Based Detection of Sexism and Source Intention in Memes Using a Learning with Disagreement Approach

Alvaro Carrillo-Casado*, Javier Román-Pásaro, Jacinto Mata-Vázquez and Victoria Pachón-Álvarez

I2C Research Group, University of Huelva, Spain

Abstract

In this paper, the I2C-UHU Group addresses the Exist-2024 challenges of Sexism Identification and Source Intention in Memes. We developed an ensemble of classifiers based on Transformer technology and adopted a Learning with Disagreement (LeWiDi) approach to analyze data from multiple annotators' perspectives. Techniques for constructing datasets and optimizing hyperparameters were explored, enhancing model performance through varied combinations. The optimal models were refined by weighting according to prediction accuracy. Our submissions for Task 4 achieved ranks of 4th with ICM-Hard and ICM-Soft scores of 0.5668 and 0.4476, respectively. For Task 5, we secured 2nd and 10th places with ICM-Hard and ICM-Soft scores of 0.4119 and 0.2023, respectively.

Keywords

Transformers, Ensemble of classifiers, Learning with Disagreement, Memes, Hyperparameter, Sexism

1. Introduction

Recent years have seen a marked increase in the prevalence of memes on social media, a distinct type of imagery characterized by humorous textual content. This study investigates how such memes can be used to entertain and disseminate sexist content. This type of humour is often utilized to harm others, for instance through sexism. However, natural language processing (NLP) is an effective tool for understanding and analysing such content.

This paper presents our research on developing a system to detect sexism and the creator's intention in memes, using natural language processing techniques as part of the tasks Sexism Identification in Memes and Source Intention in Memes of EXIST 2024 [1]. For this purpose, models based on Transformers [2] were developed, different types of dataset constructions were performed [3], followed by utilizing the Learning with Disagreement (LeWiDi) [4] approach to build models based on the various perspectives of the annotators, and finally, they were assembled to improve the performance of the models.

In Section 2, we delineate prior research efforts, while Section 3 provides a detailed exposition of Tasks 4 and 5 within the EXIST 2024 framework. Subsequently, Sections 4 and 5, expound

CLEF 2024: Conference and Labs of the Evaluation Forum, September 9–12, 2024, Grenoble, France

*Corresponding author.

✉ alvaro.carrillo121@alu.uhu.es (A. Carrillo-Casado); javier.roman780@alu.uhu.es (J. Román-Pásaro); 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).

upon the methodology employed and the resultant findings. Finally, Section 6 encapsulates the study's conclusions and outlines prospective avenues for future research endeavors.

2. Related Works

As previously indicated, one of the foundational elements employed in this study is the Learning with Disagreement (LeWiDi) approach. When information from multiple annotators was available during the classifier's creation, the decision generally favoured the majority's opinion. Nonetheless, this method could overlook valuable insights that might enhance the models' effectiveness.

In [5], participation by the AIT_FHSTP team in the EXIST2021 benchmark was noted, concentrating on the automated detection of sexism across social networks using machine learning techniques. This effort was approached as both a binary classification problem and a more detailed task that categorized various forms of sexist content. Two multilingual Transformer models were utilized for their analysis: one based on Multilingual BERT and the other on XLM-R. These models underwent adaptation through unsupervised pre-training and were subsequently fine-tuned with additional data to optimize performance.

Furthermore, in [6], irony is analyzed based on the principles of data perspectivism. It was observed how data, varying by origin, age, and gender, were managed. The performance derived from the standard test set was compared with that from a perspective-based test set. The latter detected the positive class more accurately, demonstrating the effectiveness of incorporating diverse annotator viewpoints.

3. Tasks and Dataset Description

The objective of *Task 4: Sexism Identification in Memes* is to determine which memes are sexist, while *Task 5: Source Intention in Memes* involves categorizing memes based on the author's intention to understand the role of social media in disseminating sexist messages. The dataset labels are "DIRECT," "JUDGEMENTAL," "-", and "UNKNOWN." For this study, the classification is focused on distinguishing between "DIRECT," where the intention is to spread a sexist message, and "JUDGEMENTAL," where the intention is to condemn a sexist situation or behavior. Both tasks are binary classification tasks.

The features of each meme are:

- id_EXIST : a unique identifier for the meme.
- lang : languages of the meme ("en" or "es").
- text : text automatically extracted from the meme.
- meme : name of the file that contains the meme.
- path_memes : path to the file that contains the meme.
- number_annotators : number of persons that have annotated the meme.
- annotators : a unique identifier for each of the annotators.
- gender_annotators : gender of the different annotators. Possible values are: "F" and "M", for female and male respectively.

- age_annotators : age group of the different annotators. Possible values are: 18-22, 23-45 and 46+.
- ethnicity_annotators : self-reported ethnicity of the different annotators. Possible values are: “Black or African America”, “Hispano or Latino”, “White or Caucasian”, “Multiracial”, “Asian”, “Asian Indian” and “Middle Eastern”.
- study_level_annotators : self-reported level of study achieved by the different annotators. Possible values are: “Less than high school diploma”, “High school degree or equivalent”, “Bachelor’s degree”, “Master’s degree” and “Doctorate”.
- country_annotators : self-reported country where the different annotators live in.
- labels_task4 : a set of labels (one for each of the annotators) that indicate if the meme contains sexist expressions or refers to sexist behaviours or not. Possible values are: “YES” and “NO”.
- labels_task5 : a set of labels (one for each of the annotators) recording the intention of the person who created the meme. Possible labels are: “DIRECT”, “JUDGEMENTAL”, “”, and “UNKNOWN”.
- split : subset within the dataset the meme belongs to (“TRAIN-MEME”, “TRAIN- MEME” + “EN”/“ES”).

The organizers provided only a training dataset; therefore, an 80%-20% split was performed for training and testing purposes. Furthermore, the training dataset was subdivided into 85% for training and 15% for validation. To establish an initial baseline, a single label was assigned using hard voting [7] among the labels proposed by the six annotators. Given the even number of annotators, ties were resolved by randomly selecting a label. Table 1 displays the class distribution for Task 4 following the voting process.

Table 1
Class distribution for Task 4

Class	Total	YES	NO
Train	2749	1810	939
Valid	446	331	155
Test	809	476	333

For Task 5, since only two labels (“DIRECT” and “JUDGEMENTAL”) need to be detected, a hard voting strategy was also used to generate the hard label among the annotators. The values “-” and “UNKNOWN” were discarded in the voting process. Table 2 shows the class distribution for Task 5 after the voting process.

Table 2
Class distribution for Task 5

Class	Total	DIRECT	JUDGEMENTAL
Train	2498	1668	830
Valid	440	293	147
Test	721	482	239

4. Methodology and Experiments

In this section, we delineate the methodologies employed in our investigation. Despite the availability of visual content in the provided meme datasets, our analytical approach was exclusively focused on the textual data extracted from these memes. This decision was driven by our aim to develop and refine text-based classifiers capable of effectively discerning sexism and source intentions within the content.

One of the primary innovations of this study lies in the utilization of three distinct training datasets for experimentation. Given that the data encompass two languages, English and Spanish, we employed two translation techniques to generate supplementary training datasets. For task resolution, we leveraged language models founded on Transformer architectures. Specifically, our approach entailed the utilization of two multilingual models: BERT [8] and RoBERTa [9]. The fine-tuning process of these models was meticulously optimized through a comprehensive search for optimal hyperparameter values, as elaborated in Section 4.3. The models chosen for inclusion in the study were:

- bert-base-multilingual-uncased [8]: This model is the multilingual version of BERT.
- xlm-roberta-base [10]: This model is the multilingual version of RoBERTa.

In addition to using a single hard label, we have explored and trained the models from the perspective of the annotators using various strategies, which will be described in the following sections.

To compare the results, a baseline was constructed using the two selected models with default hyperparameters: a batch size of 32, a learning rate of 3e-5, a maximum sequence length of 128, and a weight decay of 0.01. Tables 3 and 4 show the F1 score achieved by the models.

Table 3
Baselines for Task 4

Model	F1 Score
BERT	0.6395
XLM-RoBERTa	0.6626

Table 4
Baselines for Task 5

Model	F1 Score
BERT	0.5481
XLM-RoBERTa	0.5520

4.1. Data Pre-processing

Data preprocessing in this study involved an initial comprehensive processing of textual content from memes. This processing included converting all text to lowercase, and removing links, usernames, and hashtag symbols ('#'). Subsequent empirical evaluations demonstrated that additional preprocessing steps did not yield significant improvements in test outcomes. Consequently, the final preprocessing strategy was refined to include only the conversion of text to lowercase.

4.2. Dataset Construction

The dataset, as illustrated in Tables 1 and 2 comprises a constrained quantity of instances. To address this constraint, various strategies were employed to increase the amount of data, similar to those used in data augmentation. We leveraged the fact that the data provided by the organization are in both English and Spanish by translating each instance into the opposite language, thereby creating a new dataset with double the data.

The other technique employed was back-translation [11], where each instance was translated into a different language (in this case, German) and then translated back into the original language. We leveraged the accuracy of ChatGPT [12] for this process. These augmented datasets were then combined with the original dataset to create three datasets for experimentation:

- Original : The training dataset provided by the organization.
- Simple : Original plus simple translation extension.
- Back : Original plus back-translation extension.

4.3. Hyperparameter Search

Hyperparameter search [13] is one of the most important steps for model fine-tuning. Various combinations of hyperparameters were evaluated, and the number of instances was reduced to shorten experimentation time. The Optuna library [14] in Python was used, which allows us to establish the hyperparameter space to find the best ones according to a specified metric.

Table 5
Hyperparameters space

Hyperparameter	Values
Batch Size	[8, 16, 32]
Learning Rate	[1e-05, 3e-05, 5e-05]
Weight Decay	[0.01, 0.1]

Table 5 shows the hyperparameter space, and Tables 6 and 7, show the best hyperparameters for each task.

Table 6
Best hyperparameters for Task 4

Hyperparameter	BERT	XLM-RoBERTa
Batch Size	16	16
Learning Rate	3e-05	1e-05
Weight Decay	0.1	0.01

Table 7
Best hyperparameters for Task 5

Hyperparameter	BERT	XLM-RoBERTa
Batch Size	32	32
Learning Rate	3e-05	1e-05
Weight Decay	0.1	0.01

4.4. Model Perspectives

Models training based on annotators' perspectives were employed, motivated by the abundance of features available within the dataset. This approach allows using only a specific perspective or combining as many as desired, although it is computationally more expensive. In our case, the eight perspectives with the most number of examples were chosen and trained with the three datasets mentioned above to create a final model by combining all the best perspectives.

For each perspective, the data were balanced by means a undersampling technique. The selected perspectives are: gender("M", "F"), age("23-45", "18-22", "46+"), studies("Bachelor's degree", "High school degree or equivalent"), and ethnicity ("White or Caucasian").

Table 8

F1-Score results for the perspectives for Task 4

Model	Dataset	M	F	23-45	18-22	46+	Bachelor's	High school	White
BERT	Original	0.6580	0.6205	0.5920	0.6360	0.6231	0.6131	0.6270	0.6611
BERT	Simple	0.6213	0.6191	0.6255	0.6417	0.6223	0.6170	0.6150	0.6679
BERT	Back	0.6613	0.5699	0.6710	0.6507	0.5967	0.6349	0.6228	0.6393
XLM-RoBERTa	Original	0.6251	0.6264	0.6700	0.6493	0.6270	0.6559	0.6553	0.6471
XLM-RoBERTa	Simple	0.6445	0.6457	0.6744	0.6469	0.6425	0.6375	0.6387	0.6768
XLM-RoBERTa	Back	0.6468	0.6305	0.6637	0.6398	0.6299	0.6431	0.6501	0.6688

Table 9

F1-Score results for the perspectives for Task 5

Model	Dataset	M	F	23-45	18-22	46+	Bachelor's	High school	White
BERT	Normal	0.5375	0.5805	0.5609	0.5132	0.5500	0.5357	0.5402	0.5616
BERT	Simple	0.5148	0.5170	0.5324	0.4991	0.4923	0.5033	0.5222	0.5204
BERT	Back	0.5225	0.5159	0.5314	0.5460	0.5649	0.5516	0.5095	0.5567
XLM-RoBERTa	Normal	0.5322	0.5275	0.5474	0.5311	0.4906	0.5317	0.5372	0.5738
XLM-RoBERTa	Simple	0.5472	0.4938	0.5572	0.5517	0.5589	0.5502	0.5390	0.5635
XLM-RoBERTa	Back	0.5299	0.5336	0.5449	0.5591	0.5760	0.5432	0.5479	0.5072

In Tables 8 and 9, the selected models for each perspective are highlighted. Given our approach of treating the models separately, we choose the best model for each perspective based on the dataset employed for its training. For example, the Model 1 is composed of perspective "M" with the training dataset "Back", "F" with "Original", "23-45" with "Back", "18-22" with "Back", "46+" with "Original", "Bachelor's" with "Back", "High school" with "Original" and "White" with "Simple". The architecture of our ensemble models is structured as follows:

- Model 1 and Model 4: More efficient BERT models from each perspective for Task 4 and Task 5 respectively.
- Model 2 and Model 5: More efficient XLM-RoBERTa models from each perspective for Task 4 and Task 5 respectively.
- Model 3 and Model 6: More efficient BERT/XLM-RoBERTa models from each perspective for Task 4 and Task 5 respectively.

4.5. Ensemble Approach

This section describes our ensemble approach to obtain a single prediction based on the predictions obtained individually from each perspective. This strategy involves assigning a weight to each individual prediction through a joint weight search process to obtain overall F1.

Table 10
Weight values space

Weights
$\{0.5, 0.75, 1, 1.25, 1.5, 1.75\}$

In Table 10, the possible weight values assigned to the predictions of each perspective are displayed.

Table 11
Final combination for Task 4

Number	Model	M	F	23-45	18-22	46+	Bachelor's	High school	White	Overall F1 score
1	Model 1	1.25	0.75	1.25	0.75	0.5	0.5	1.5	1.5	0.7294
2	Model 1	1.25	0.75	1.25	0.75	0.5	0.5	1.25	1.75	0.7288
3	Model 2	1.75	0.5	1.5	0.75	0.75	0.5	0.5	1.75	0.7052
4	Model 2	1.75	0.5	1.25	0.75	0.75	0.5	0.75	1.75	0.7029
5	Model 3	0.75	0.5	1.75	1.75	0.5	1	1	0.75	0.7224
6	Model 3	1.5	1	0.75	1.75	0.5	0.5	1.5	0.5	0.7187

Table 12
Final combination for Task 5

Number	Model	M	F	23-45	18-22	46+	Bachelor's	High school	White	Overall F1 score
7	Model 4	1.25	0.5	1.75	0.5	1	1	1.5	0.5	0.6352
8	Model 4	1	0.5	1.75	0.5	1	1	1.75	0.5	0.6329
9	Model 5	0.5	1	1.75	0.75	1.75	0.75	0.5	1	0.6061
10	Model 5	0.5	1	1.5	1	1.5	0.75	0.5	1.25	0.6049
11	Model 6	1.5	1.5	1.5	0.75	1.25	0.5	0.5	0.5	0.6147
12	Model 6	1.75	1.25	1.5	0.5	1	0.5	1	0.5	0.6114

As observed in Tables 11 and 12, the approach based on training models using annotators' perspectives and the weight-based ensemble significantly improve the results over the baselines shown in Tables 3 and 4, respectively. The three best models for Task 4 found Table in 11 are:

- Run 1 (I2C-Huelva_1): Model 1 with balanced weights number 1.
- Run 2 (I2C-Huelva_2): Model 1 with balanced weights number 2.
- Run 3 (I2C-Huelva_3): Model 3 with balanced weights number 5.

For Task 5, a run from Task 4 was chosen and its result was evaluated with the following models in Table 12:

- Run 4 (I2C-Huelva_1) : Run 1 with Model 4 and balanced weights number 7.
- Run 5 (I2C-Huelva_2) : Run 3 with Model 4 and balanced weights number 7.
- Run 6 (I2C-Huelva_3) : Run 2 with Model 4 and balanced weights number 8.

4.6. Error Analysis

In this section, the errors of the models will be examined through the analysis of their confusion matrices. This approach will allow a detailed understanding of the models' performance, identifying both their successes and failures in classifying the samples. This critical evaluation will provide valuable information for improving the accuracy and reliability of the models, thereby contributing to the advancement of the field of study.

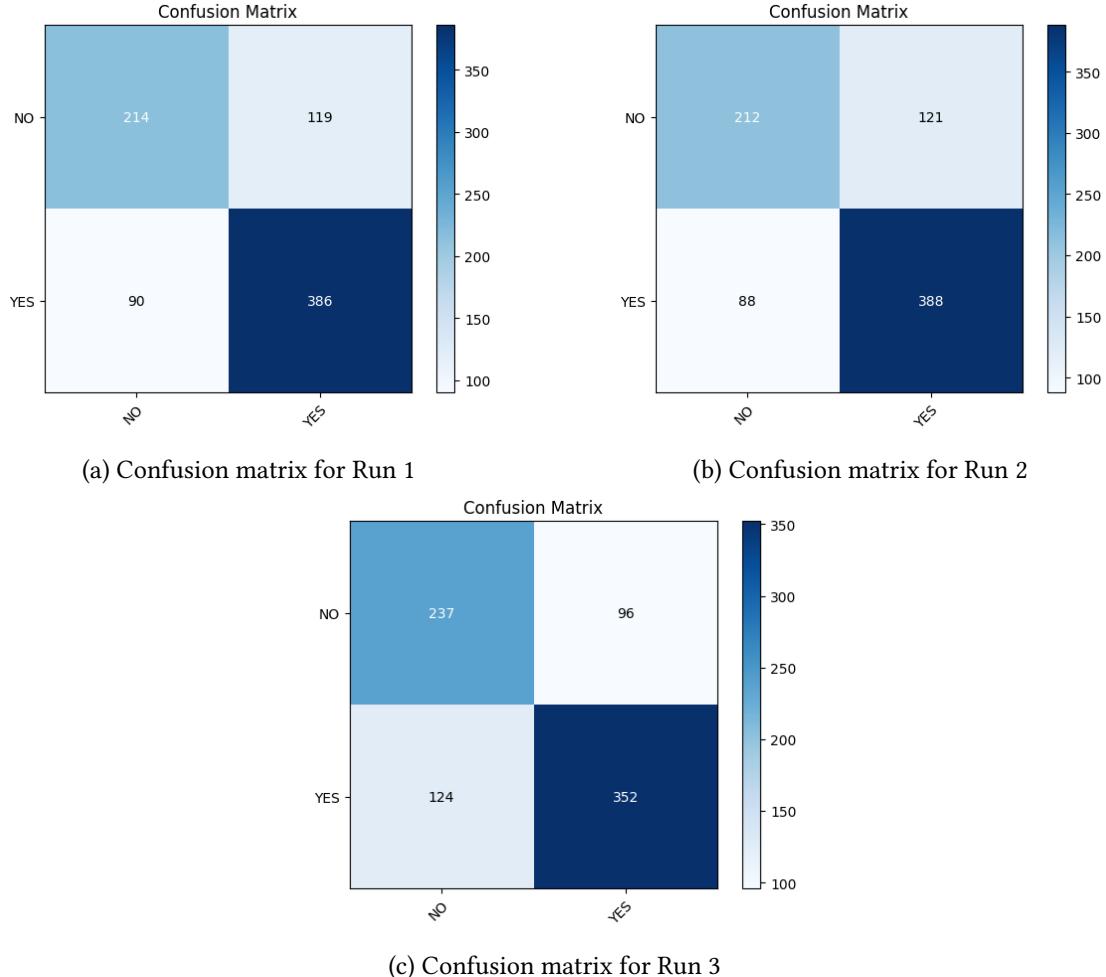


Figure 1: Confusion matrices for Task 4

For Task 4, all figures show similar overall performance patterns, with a notably high proportion of correct predictions (TP and TN) compared to incorrect ones (FP and FN). This suggests a consistent ability of the models to accurately classify samples from both positive and negative classes. However, differences between the models reveal distinct trends. Figure 1c exhibits a slightly higher number of true positives (TP) compared to Figures 1a and 1b, indicating a potentially better capability of the mixed BERT/XLM-RoBERTa model to identify positive class

samples. Conversely, Figures 1a and 1b demonstrate similar trends in false positives (FP) and false negatives (FN), while Figure 1c shows a slightly higher proportion of false negatives (FN). These discrepancies could stem from variations in model architectures (solely BERT vs. mixed BERT/XLM-RoBERTa) and the specific characteristics of the dataset and training processes. Combining these observations may inform future research on model selection and optimization for specific classification tasks.

Table 13
Examples labeled for Task 4

Text	Test	Run 1	Run 2	Run 3
metro q estilo de vida alexa recomienda a una madre asesinar a sus hijos: amazon pide disculpas por "error en la configuración" el asistente inteligente ofreció una respuesta polémica cuando una mujer le preguntó sobre "cómo evitar que los niños rían" alexa le pusieron ese nombre por no llamarla skynet más en cuantarazon.com	0	1	1	1
ME DIJO QUE ME FUERA A FREGAR memegenerator.es	1	0	0	0

Table 13 illustrates the difficulty in classifying certain texts accurately. In the first example, the text addresses controversial topics and specific entities (e.g., Amazon, Alexa), which can lead to misclassification due to lack of context. In the second example, it demonstrates the variety of topics and the presence of humorous elements that can complicate the task of automated classification.

For Task 5, Run 4 and 5 are identical, whereas Run 6 is based on the same model but with different weights used for prediction. Therefore, we will only compare the confusion matrices of Runs 4 and 6.

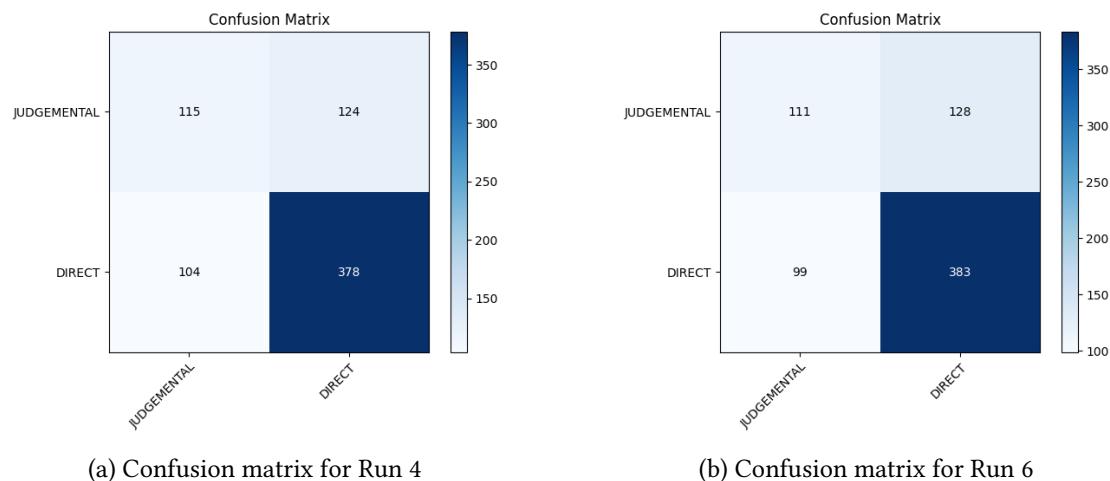


Figure 2: Confusion matrices for Task 5

Both Figures 2a and 2b are based on the same BERT architecture. The differences in error

distribution and the final model’s value system suggest that the models have been trained slightly differently. However, they share fundamental similarities due to their common foundation in BERT and their identical matrix structure.

Table 14
Examples labeled for Task 5

Text	Test	Run 4	Run 6
La mecánica es solo para hombres, toma mi bolso, sé más que tú de motores, putito.	0	1	1
yet men don't have the same rights as women like the right to share their opinion on abortion	1	0	0

In the first example of Table 14 the misclassification of this text could be attributed to the lack of consideration for cultural, social, and linguistic context, as well as the incapacity of an automated algorithm to capture nuances in tone and communicative intent. In the second example, the text discusses gender rights with a specific focus on the disparity in opinions on abortion, which introduces sensitive and context-dependent themes. These cases demonstrate the challenge of classifying texts with similar vocabulary but different contexts or fragmented and disjointed content.

5. Results

This section presents the results obtained from the competition, detailing the performance of our top submissions across various tasks. The metrics to be evaluated for the competition are:

- Hard-Hard: The ‘hard’ labels are derived from the annotators’ labels using probabilistic thresholds specific to each task.
 - Task 4: The class annotated by more than 3 annotators is selected.
 - Task 5: The class annotated by more than 2 annotators is selected.

Items without a majority class are removed from the evaluation. The official metric is the original ICM, and F1 (the harmonic mean of precision and recall) is also used for comparison.

- Soft-Soft: Compares the probabilities assigned by the system with those assigned by the human annotators. As in the previous case, ICM-soft will be used as the official evaluation metric.

Our final models returned a percentage corresponding to the Soft-Soft measure. For the Hard-Hard measure, it was filtered if that percentage was greater than 50%. Tables 15 to 18 show the official results obtained by the submitted runs.

Table 15

Ranking of participants for Task 4 Hard-Hard

Rank	Run	ICM-Hard	ICM-Hard Norm	F1_YES
1	RoJiNG-CL_3	0.3182	0.6618	0.7642
2	RoJiNG-CL_2	0.2272	0.6155	0.7437
3	RoJiNG-CL_1	0.1863	0.5947	0.7274
4	I2C-Huelva_2	0.1313	0.5668	0.7241
5	I2C-Huelva_1	0.1166	0.5593	0.7154
-	-	-	-	-
9	I2C-Huelva_3	0.0987	0.5502	0.6933
-	-	-	-	-
53	melialo-vcassan_1	-0.8109	0.0876	0.5316

Table 16

Ranking of participants for Task 4 Soft-Soft

Rank	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
1	Victor-UNED_1	-0.2925	0.4530	1.1028
2	Victor-UNED_2	-0.3135	0.4496	1.2834
3	Elias&Sergio_1	-0.3225	0.4482	0.9903
4	I2C-Huelva_3	-0.3263	0.4476	1.5189
5	I2C-Huelva_1	-0.3390	0.4455	1.4096
6	I2C-Huelva_2	-0.3446	0.4446	1.4112
-	-	-	-	-
37	CNLP-NITS-PP_1	-2.6987	0.0662	1.3445

Table 17

Ranking of participants for Task 5 Hard-Hard

Rank	Run	ICM-Hard	ICM-Hard Norm	Macro F1
1	Victor-UNED_1	-0.2397	0.4167	0.3873
2	I2C-Huelva_2	-0.2535	0.4119	0.4761
3	Victor-UNED_2	-0.2668	0.4073	0.3850
4	I2C-Huelva_3	-0.2772	0.4036	0.4714
5	I2C-Huelva_1	-0.2880	0.3999	0.4714
-	-	-	-	-
22	epistemologos_1	-8.7012	0.0000	0.0557

Table 18

Ranking of participants for Task 5 Soft-Soft

Rank	Run	ICM-Soft	ICM-Soft Norm	Cross Entropy
1	Victor-UNED_2	-1.2453	0.3676	1.6235
-	-	-	-	-
8	melialo-vcassan_3	-2.0653	0.2804	1.5295
9	melialo-vcassan_1	-2.6821	0.2148	1.6291
10	I2C-Huelva_3	-2.7996	0.2023	3.9604
11	I2C-Huelva_2	-2.7997	0.2023	3.9857
12	I2C-Huelva_1	-2.8007	0.2022	3.9735
-	-	-	-	-
17	Penta-ML_2	-5.9832	0.0000	5.4845

6. Conclusions and Future Works

In this study, the identification of sexism and source intentions in memes was explored, and the findings were presented at the EXIST 2024 competition. Various methodologies were evaluated to develop the most effective classifiers, employing both conventional models based on hard voting and innovative models utilizing the Learning with Disagreement (LeWiDi) approach. It was found that the latter approach, which incorporates perspectives from diverse annotators, exhibited superior performance compared to the traditional models. Consequently, notable rankings were achieved: fourth place was secured in both the Hard-Hard and Soft-Soft measures for Task 4, and second and tenth places were obtained for Task 5, respectively.

Looking forward, the methodologies applied in this research are planned to be refined, and the focus is intended to be expanded to include image analysis. This enhancement aims to develop a more comprehensive model that integrates visual elements with textual analysis, thereby advancing the capability to detect sexist content in memes.

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 Alboroz, 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: European Conference on Information Retrieval, Springer, 2024, pp. 498–504.
- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

- [3] Y. Li, X. Li, Y. Yang, R. Dong, A diverse data augmentation strategy for low-resource neural machine translation, *Information* 11 (2020) 255.
- [4] A. N. Uma, T. Fornaciari, D. Hovy, S. Paun, B. Plank, M. Poesio, Learning from disagreement: A survey, *Journal of Artificial Intelligence Research* 72 (2021) 1385–1470.
- [5] M. Schütz, J. Boeck, D. Liakhovets, D. Slijepčević, A. Kirchknopf, M. Hecht, J. Bogensperger, S. Schlarb, A. Schindler, M. Zeppelzauer, Automatic sexism detection with multilingual transformer models, *arXiv preprint arXiv:2106.04908* (2021).
- [6] S. Frenda, A. Pedrani, V. Basile, S. M. Lo, A. T. Cignarella, R. Panizzon, C. Sánchez-Marco, B. Scarlini, V. Patti, C. Bosco, et al., Epic: Multi-perspective annotation of a corpus of irony, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023, pp. 13844–13857.
- [7] D. M. Tax, M. Van Breukelen, R. P. Duin, J. Kittler, Combining multiple classifiers by averaging or by multiplying?, *Pattern recognition* 33 (2000) 1475–1485.
- [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] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, *arXiv preprint arXiv:1907.11692* (2019).
- [10] A. Conneau, K. Khadwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, *arXiv preprint arXiv:1911.02116* (2019).
- [11] D. R. Beddiar, M. S. Jahan, M. Oussalah, Data expansion using back translation and paraphrasing for hate speech detection, *Online Social Networks and Media* 24 (2021) 100153.
- [12] Y. Gao, R. Wang, F. Hou, How to design translation prompts for chatgpt: An empirical study, *arXiv e-prints* (2023) arXiv–2304.
- [13] L. Yang, A. Shami, On hyperparameter optimization of machine learning algorithms: Theory and practice, *Neurocomputing* 415 (2020) 295–316.
- [14] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.