



**UNIVERSIDAD DE HUELVA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA**

**Técnicas de Deep Learning para Identificar y Clasificar
Mensajes de Odio en Redes Sociales Dirigidos a la
Comunidad LGTBQ+**

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Autores: Antonio José Morano Moriña y Javier Román Pásaro

Tutor(a): Jacinto Mata Vázquez

Co-tutor(a): Victoria Pachón Álvarez

septiembre, 2023

Técnicas de Deep Learning para Identificar y Clasificar Mensajes de Odio en Redes Sociales Dirigidos a la Comunidad LGBTQ+

© Antonio José Morano Moriña y Javier Román Pásaro, 2023

Este documento se distribuye con licencia CC BY-NC-SA 4.0. El texto completo de la licencia puede obtenerse en <https://creativecommons.org/licenses/by-nc-sa/4.0/>.

La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso, sean preservados en todas las copias.



Técnicas de Deep Learning para Identificar y Clasificar Mensajes de Odio en Redes Sociales Dirigidos a la Comunidad LGBTQ+

Antonio José Morano Moriña y Javier Román Pásaro
Huelva, septiembre 2023

Resumen

En la actualidad, en el entorno digital, el procesamiento del lenguaje natural (PLN) ha adquirido una relevancia crucial como disciplina para comprender y analizar la gran cantidad de información generada en las plataformas de redes sociales. La capacidad de extraer conocimientos significativos de datos textuales resulta fundamental en diversos campos, como la investigación social, la toma de decisiones políticas y la detección de problemas sociales. En este contexto, la detección de comentarios fóbicos dirigidos hacia la comunidad LGBTQ+ ha ganado importancia debido a la necesidad de fomentar la inclusión, el respeto y la igualdad en línea.

En el marco de IberLef-2023, se ha presentado un enfoque para abordar la tarea denominada “HOMO-MEX: la detección de discurso de odio en mensajes en línea dirigidos hacia la población LGBTQ+ de habla hispana en México”. La contribución principal radica en la demostración de la eficacia de utilizar un conjunto de clasificadores basados en transformers. Mediante la combinación de múltiples modelos, se lograron aprovechar las fortalezas individuales de cada uno, obteniendo así un rendimiento mejorado en comparación con el uso de un único modelo. La tarea mencionada comprendía dos subtareas: clasificación multiclase (clases P, NP y NA) y clasificación multietiqueta (etiquetas G, L, B, T y O). En el caso de la clasificación multietiqueta, se implementaron cinco clasificadores binarios independientes, uno por cada etiqueta.

Además de abordar esta nueva tecnología, se aplicaron diversos métodos para modificar los datos y mejorar los resultados. El primer objetivo fue eliminar el ruido de las instancias mediante la normalización y la aplicación de capas de preprocesamiento, como la eliminación de menciones, URLs o emoticonos. Para equilibrar los conjuntos de datos, se empleó la técnica de back-translation sobre las instancias de la clase minoritaria (positiva), lo que permitió obtener más instancias positivas y, por ende, un conjunto de datos más balanceado. Este método se basa en traducir el texto de los tweets a un idioma intermedio para luego volver a traducirlo al idioma original, generando así otra instancia con el mismo significado semántico, pero con una forma y estructura gramatical diferente.

Un aspecto destacado es la importancia de seleccionar los hiperparámetros adecuados durante el proceso de entrenamiento del modelo. Para lograrlo, se hizo uso de la plataforma WanDB para entrenar los modelos con un conjunto de hiperparámetros y seleccionar aquellas combinaciones que ofrecían los mejores resultados en las predicciones. Finalmente, se probaron varios modelos y se decidió combinar los tres que proporcionaron el mejor valor de F1 para el conjunto de datos en cuestión.

Este enfoque meticuloso y los resultados obtenidos contribuyen a promover la inclusión, el respeto y la igualdad en línea, y subrayan la necesidad continua de desarrollar enfoques efectivos en el procesamiento del lenguaje natural para abordar los desafíos actuales en las redes sociales.

Palabras clave: Comunidad LGBTQ+, PLN, Deep Learning, Transformers, Hiperparámetro, Data Augmentation, Ensamblador

I2C-UHU at IberLEF-2023 HOMO-MEX task: Ensembling Transformers Models to Identify and Classify Hate Messages Towards the Community LGBTQ+

Antonio José Morano Moraña y Javier Román Pásaro
Huelva, September 2023

Abstract

This paper presents the approaches proposed for I2C Group to address the IberLef-2023 Task HOMO-MEX: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGTBQ+ population. The major contribution has been the demonstration of the effectiveness of using an ensemble of classifiers based on transformers. By combining multiple models, the individual strengths were leveraged, resulting in improved performance compared to using a single model. Furthermore, the significance of selecting appropriate hyperparameters during the model training process was underscored by the results. Through meticulous experimentation and evaluation of different hyperparameter combinations, the settings that reached the best performance for the given tasks were identified. In our experiments for both tasks we have tested several models and decided to ensemble the three models that provided the best F1-Score for this dataset. Additionally, for Task 2 we decided to train individual binary classifiers for each class instead of making a multilabel classifier. The model submitted for Task 1 achieved a F1-Score of 83,25%, ranking in the 6th place of the competition. The model for the Task 2 reached a F1-Score of 69,60%, ranking in the 1st place of the competition.

Keywords: Deep Learning, Transformers, Ensembler, Hyperparameter, Twitter, LGBT-Phobia, Hate Speech Detection

Agradecimientos

Primero de todo, agradecer a mis tutores, sin los cuales no existiría este proyecto y a todos aquellos docentes que me han impulsado a mejorar a lo largo de mi vida.

A mi amigo Javier, por el trabajo realizado en el proyecto y los años de amistad.

A mis padres, por quererme, guiarme y apoyarme incondicionalmente en cada paso que doy.

A mi hermano, por compartir recuerdos, sueños e incontables experiencias.

Antonio José Morano Moriña
Huelva, 2023

En primer lugar, me gustaría darle las gracias a todos los docentes que he tenido a lo largo de mi vida estudiantil, por ayudarme y crearme la necesidad de querer aprender. A mis tutores por la dedicación y la enseñanza.

También a mi compañero Antonio, por la paciencia y el esfuerzo depositado en el proyecto.

Gracias a mi familia. A mi padre por ser el ejemplo en el que fijarme, a mi madre por el apoyo incondicional y a mi hermano por el amor fraternal.

Especialmente gracias a Laura, por ser el faro en los momentos difíciles y mi hogar cuando todo parece perdido.

Javier Román Pásaro
Huelva, 2023

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
Índice de figuras	xiii
Índice de tablas	xv
1. Introducción	1
1.1. Objetivos del proyecto	1
1.2. Estructura del documento	1
2. Marco Teórico	3
2.1. Clasificación automática de textos	3
2.2. Word embeddings	7
2.3. Conjunto de datos desbalanceados	8
3. Desarrollo y Resultados	11
3.1. Descripción de la tarea	11
3.2. Metodología	14
3.3. Resultados	19
3.4. Análisis de errores	20
4. Propuestas de mejora	23
4.1. Técnicas de balanceo	23
4.2. Técnicas de generación de datos sintéticos	24
5. Conclusiones y trabajos futuros	27
5.1. Conclusiones	27
5.2. Trabajos futuros	28
Bibliografía	29
A. Working Notes	33

Índice de figuras

2.1.	Estructura de una red neuronal	5
2.2.	Estructura de una red neuronal recurrente	6
2.3.	Arquitectura de los Transformer	7
2.4.	Ejemplo de Back-translation	9
2.5.	Técnicas de balanceo, undersampling y oversampling	10
3.1.	Matriz de confusión para la Subtarea 1	21
3.2.	Matrices de confusión para la Subtarea 2	22
4.1.	Estudio de las proporciones en el undersampling	24

Índice de tablas

3.1.	Ejemplo de instancias de la Subtarea 1	12
3.2.	Distribución de clases de la Subtarea 1	13
3.3.	Ejemplo de instancias de la Subtarea 2	13
3.4.	Distribución de clases de la Subtarea 2	13
3.5.	Estructura de una matriz de confusión	14
3.6.	Resultados del Baseline en la Subtarea 1	15
3.7.	Resultados del Baseline en la Subtarea 2	15
3.8.	Aplicación del preprocesamiento	16
3.9.	Resultados tras el Preprocesamiento en la Subtarea 1	16
3.10.	Resultados tras el Preprocesamiento en la Subtarea 2	16
3.11.	Aplicación del back-translation	17
3.12.	Espacio de hiperparámetros	18
3.13.	Mejores hiperparámetros por modelo	18
3.14.	Resultados tras data augmentation y búsqueda de hiperparámetros en la Subtarea 1	18
3.15.	Resultados tras data augmentation y búsqueda de hiperparámetros en la Subtarea 2	19
3.16.	Ejemplo del uso de la técnica de ensemble sobre una instancia de la Subtarea 2	19
3.17.	Resultados finales en la Subtarea 1	19
3.18.	Resultados finales en la Subtarea 2	20
3.19.	Ranking de participantes de la Subtarea 1	20
3.20.	Ranking de participantes de la Subtarea 2	20
3.21.	Análisis de errores de la Subtarea 1	22
4.1.	Nuevas proporciones tras el uso del undersampling	23
4.2.	Resultados tras undersampling en la Subtarea 1	24
4.3.	Resultados tras bootstrap en la Subtarea 1	25
4.4.	Resultados tras bootstrap en la Subtarea 2	25

CAPÍTULO 1

Introducción

En la era digital actual, el procesamiento del lenguaje natural (PLN) [1], concretamente, la habilidad de extraer conocimiento valioso de datos textuales es esencial en diversas áreas, como la investigación social, la formulación de políticas y la identificación de cuestiones sociales. En este contexto, la identificación de comentarios que promueven la intolerancia hacia la comunidad LGTBQ+ ha adquirido una creciente relevancia, ya que es crucial para fomentar la inclusión, el respeto y la igualdad en el entorno digital.

En este documento se presenta la investigación sobre el desarrollo de un sistema para detectar comentarios fóbicos hacia la comunidad LGTBQ+ utilizando técnicas de procesamiento del lenguaje natural como parte de la tarea HOMO-MEX: Detección de discurso de odio hacia la población LGTBQ+ de habla hispana de México en IberLEF 2023 [2] [3]. Dado el éxito y la popularidad de los modelos Transformers, todos los modelos desarrollados se basan en esta tecnología.

1.1. OBJETIVOS DEL PROYECTO

El principal objetivo de este proyecto es investigar sobre el aprendizaje automático, Deep Learning y su aplicación en el procesamiento del lenguaje natural (PLN). En este caso, se profundizará en los Transformers [4] con el fin de conseguir los mejores resultados.

El proyecto se fundamenta en base a la competición creada por IberLef “HOMOMEX: Detección de discurso de odio en mensajes en línea dirigidos hacia la población LGTBQ+ de habla hispana en México”.

IberLEF es una organización enfocada en la investigación sobre PLN basada en competiciones internacionales. Esta publica ciertos problemas con temas bastante diversos con el objetivo de ampliar y orientar la investigación sobre el procesamiento del texto.

1.2. ESTRUCTURA DEL DOCUMENTO

En el capítulo 2 se describen teóricamente las técnicas aplicadas para la resolución de la tarea. Se presentan técnicas de clasificación de textos, con conceptos de la tecnología utilizada (Transformers) y métodos de balanceo de datos.

En el capítulo 3 se explican las técnicas aplicadas para el desarrollo de la tarea. También se muestran y analizan los resultados obtenidos, así como datos importantes para la resolución de las tareas.

En el capítulo 4 se proponen diversas opciones de mejora, centradas en el balanceo de datos, las cuales se han implementado para tratar de superar los resultados obtenidos en la competición.

En el capítulo 5 se exponen las conclusiones obtenidas a lo largo del transcurso del proyecto, como nuevos conocimientos y posibles usos. Además, se enumeran posibles casos de estudio de futuros trabajos, para el preprocesamiento y el balanceo de los datos.

Finalmente en el anexo se adjunta el paper científico derivado de la participación en iberLef, aceptado y publicado en las actas del congreso

Marco Teórico

En este punto se describen los conceptos teóricos necesarios para saber cómo se ha ido abordando la tarea en la que se ha participado, la cual abarca desde algoritmos de Machine Learning clásicos como las Redes Bayesianas hasta técnicas de Deep Learning como los Transformers.

2.1. CLASIFICACIÓN AUTOMÁTICA DE TEXTOS

2.1.1. Técnicas de Machine Learning

El Machine Learning [5] [6], también conocido como aprendizaje automático, es una rama de la inteligencia artificial que permite a las computadoras aprender sin programación explícita. En lugar de seguir un conjunto de instrucciones estáticas, los algoritmos de Machine Learning utilizan datos para mejorar su desempeño en tareas específicas. Este campo ha experimentado un rápidocrecimiento debido a los avances tecnológicos y la abundancia de datos.

Existen tres tipos principales de aprendizaje automático [7]:

Aprendizaje supervisado: Los algoritmos utilizan un conjunto de datos etiquetados que contiene ejemplos de entrada y salida esperada para aprender a hacer predicciones precisas sobre nuevas instancias no vistas previamente.

Aprendizaje no supervisado: Los algoritmos se enfrentan a un conjunto de datos sin etiquetas y buscan patrones y estructuras ocultas, como agrupaciones o reducción de dimensiones, sin una guía específica.

Aprendizaje por refuerzo: El modelo aprende a través de la interacción con un entorno y busca maximizar las recompensas a lo largo del tiempo, lo que conduce a un comportamiento más eficiente y óptimo.

El proceso de Machine Learning generalmente involucra las siguientes etapas:

Recopilación de datos: Se recopilan y preparan los datos relevantes para el problema a resolver, asegurando su calidad y cantidad.

Preprocesamiento de datos: Los datos se limpian, normalizan y transforman para eliminar ruidos y hacerlos adecuados para el algoritmo seleccionado.

Selección del modelo: Se elige el algoritmo de Machine Learning más apropiado para el problema y se ajustan sus parámetros.

Entrenamiento del modelo: Se utilizan datos de entrenamiento para ajustar los parámetros del modelo y mejorar su rendimiento.

Evaluación del modelo: Se emplea un conjunto de datos de prueba para evaluar el rendimiento del modelo y asegurar su capacidad de generalización a nuevas instancias.

El Machine Learning se aplica en diversas áreas, como reconocimiento de patrones, procesamiento del lenguaje natural, visión por computadora, recomendación de productos, diagnósticos médicos, pronósticos financieros y más. Sin embargo, es fundamental comprender sus limitaciones y considerar aspectos éticos y de privacidad al utilizarlo en aplicaciones del mundo real.

2.1.2. Redes neuronales

Las redes neuronales [8] son modelos de aprendizaje automático inspirados en la estructura y el funcionamiento del cerebro humano. Están compuestas por unidades llamadas neuronas artificiales, organizadas en capas y conectadas mediante conexiones ponderadas para permitir la transmisión de información a través de la red.

El proceso de entrenamiento de una red neuronal implica ajustar los pesos de las conexiones para que el modelo pueda aprender a hacer predicciones precisas en tareas específicas, como clasificación o regresión [9]. Esto se logra utilizando un conjunto de datos de entrenamiento con ejemplos etiquetados para actualizar los pesos y reducir el error entre las predicciones y las etiquetas reales.

En el funcionamiento de las redes neuronales, las matemáticas desempeñan un papel fundamental. Entre estos conceptos podríamos destacar la función de activación, la cual está presente en todas las neuronas artificiales, que le permite aprender relaciones complejas y patrones en los datos, las más comunes son la función sigmoide, ReLU y tangente hiperbólica.

La propagación hacia adelante es un algoritmo destacable, donde la red propaga la entrada a través de las capas utilizando operaciones matriciales y funciones de activación para generar predicciones basadas en los pesos aprendidos.

La finalidad del entrenamiento es minimizar la función de pérdida, la cual se utiliza para medir el error entre las predicciones y las etiquetas reales, utilizando algoritmos de optimización como el descenso de gradiente, que ajusta los pesos de las conexiones en función de las derivadas de la función de pérdida.

Las redes neuronales han demostrado un rendimiento impresionante en diversas aplicaciones, desde procesamiento del lenguaje natural hasta visión por computador y reconocimiento de patrones. Su capacidad para aprender representaciones complejas y no lineales a partir de datos, combinada con avances en hardware y algoritmos, ha impulsado significativamente el campo del aprendizaje automático y ha contribuido a avances importantes en la inteligencia artificial.

2.1.3. Deep Learning

El Deep Learning [10], una rama del aprendizaje automático (Machine Learning), ha emergido como una tecnología fundamental en el ámbito de la inteligencia artificial. Esta disciplina se enfoca en entrenar redes neuronales

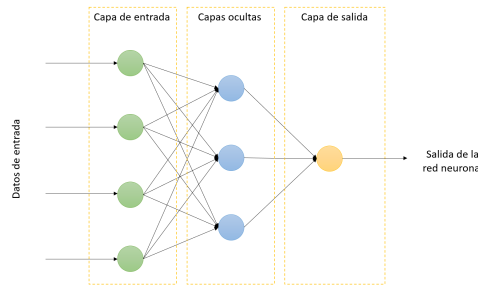


Figura 2.1: Estructura de una red neuronal

artificiales, inspiradas en el funcionamiento del cerebro humano, para abordar tareas complejas en una variedad de dominios, como visión por computadora, procesamiento del lenguaje natural y reconocimiento de patrones.

Las redes neuronales profundas constan de múltiples capas, lo que permite que el modelo adquiera representaciones complejas y de alto nivel de los datos. Cada capa contiene neuronas artificiales que procesan información y transmiten señales a través de conexiones ponderadas. Durante el entrenamiento, estos pesos de conexión se ajustan para mejorar la precisión de las predicciones.

El Deep Learning presenta características y ventajas clave, tales como:

Capacidad de aprendizaje automático de características: A diferencia de los enfoques tradicionales, el Deep Learning puede aprender automáticamente características relevantes de los datos, eliminando la necesidad de extracción manual.

Escalabilidad y adaptabilidad: Las redes neuronales profundas son adecuadas para manejar grandes conjuntos de datos y se pueden adaptar a diversas tareas, lo que las hace versátiles en la investigación.

Mejora del rendimiento con más datos: A medida que los conjuntos de datos crecen, los modelos de Deep Learning tienden a mejorar su rendimiento y capacidad de generalización, lo que es especialmente útil en entornos de Big Data.

Estado del arte en varias tareas: El Deep Learning ha superado a los enfoques tradicionales en una amplia gama de tareas, incluyendo clasificación de imágenes, reconocimiento de voz y traducción automática, logrando resultados destacados en competencias.

Sin embargo, el Deep Learning enfrenta desafíos, como la necesidad de grandes cantidades de datos etiquetados para el entrenamiento y el riesgo de sobreajuste en conjuntos de datos pequeños. A pesar de estas limitaciones, ha revolucionado el campo del aprendizaje automático al permitir que las máquinas aprendan representaciones complejas y no lineales de datos, lo que ha impulsado avances significativos en la inteligencia artificial y se ha consolidado como una herramienta poderosa para resolver problemas complejos en diversas áreas [11].

2.1.4. 2.1.4 Transformers

El Transformer [12] es una arquitectura de red neuronal diseñada para resolver problemas en el procesamiento del lenguaje natural (PLN), particularmente en tareas de traducción automática [13]. Fue introducido por Ashish Vaswani et al. en 2017 y ha transformado el campo del PLN debido a su eficiencia y rendimiento sobresaliente.

El Transformer se destaca por algunas características clave:

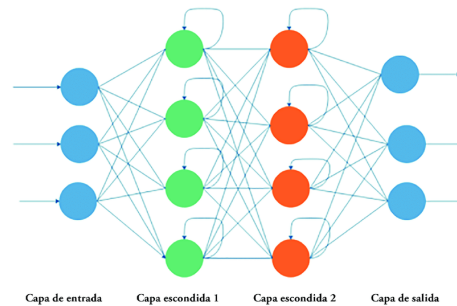


Figura 2.2: Estructura de una red neuronal recurrente

Mecanismo de atención: Se podría decir que es el corazón del Transformer, permitiendo que el modelo se enfoque en partes relevantes de una secuencia de palabras durante el procesamiento. Este mecanismo se basa en cálculos de productos escalares entre vectores de consulta, clave y valor de cada palabra, lo que habilita al modelo para aprender relaciones complejas y dependencias entre palabras.

Arquitectura basada en encoders y decoders: El Transformer consta de múltiples capas de encoders y decoders. Los encoders procesan la secuencia de entrada, mientras que los decoders se utilizan para generar secuencias de salida en tareas como la traducción automática.

Sin recurrencia ni convoluciones: A diferencia de arquitecturas anteriores como las redes neuronales recurrentes (RNN) y las redes neuronales convolucionales (CNN), el Transformer no requiere recurrencia ni convoluciones para procesar secuencias, lo que lo hace altamente paralelizable y eficiente para entrenar en unidades de procesamiento gráfico (GPU).

Los beneficios que representan el uso del Transformer son varios, entre ellos cabe destacar:

Captura de dependencias a largo plazo: Gracias a su mecanismo de atención, el Transformer puede capturar dependencias a largo plazo en secuencias, lo que mejora significativamente la calidad de las traducciones y otras tareas de PLN.

Escalabilidad: El Transformer es altamente escalable y puede manejar conjuntos de datos más grandes y tareas más complejas, lo que lo hace adecuado para aplicaciones del mundo real.

Representaciones contextuales: Al procesar la secuencia completa simultáneamente, el Transformer puede capturar el contexto global de las palabras y, por lo tanto, generar representaciones vectoriales más contextualizadas.

El Transformer se utiliza ampliamente en diversas tareas de PLN, como traducción automática, generación de texto, resumen de texto, reconocimiento de entidades, chatbots y más. Variantes del modelo original, como BERT (Bidirectional Encoder Representations from Transformers) y GPT (Generative Pre-trained Transformer), han sido entrenadas en grandes conjuntos de datos para preentrenamiento de representaciones de lenguaje, lo que ha llevado a un rendimiento aún mayor en el PLN.

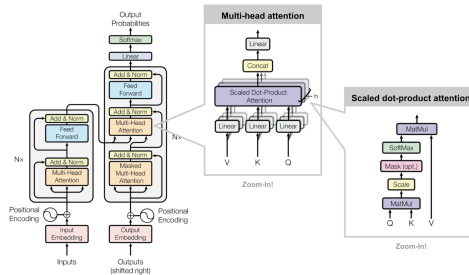


Figura 2.3: Arquitectura de los Transformer

2.2. WORD EMBEDDINGS

2.2.1. Word2Vec

Word2Vec [14] es un algoritmo popular de aprendizaje no supervisado utilizado para representar palabras en forma de vectores densos en un espacio vectorial de baja dimensión. Fue propuesto por Tomas Mikolov et al. en 2013. El objetivo principal de Word2Vec es capturar la semántica y el contexto de las palabras basándose en su distribución en grandes corpus de texto. Existen dos enfoques principales en Word2Vec: el modelo Skip-Gram y el modelo Continuous Bag of Words (CBOW). Ambos utilizan redes neuronales para aprender las representaciones vectoriales, y se basan en la idea de predecir las palabras circundantes (en el caso de Skip-Gram) o una palabra objetivo dada su(s) palabra(s) vecina(s) (en el caso de CBOW). La ventaja de Word2Vec es que las palabras con significados y contextos similares tienen representaciones vectoriales cercanas en el espacio vectorial, lo que facilita el análisis semántico y el cálculo de similitudes entre palabras.

2.2.2. Glove

GloVe (Global Vectors for Word Representation) [15] es un algoritmo de aprendizaje no supervisado para generar representaciones vectoriales (word embeddings) de palabras en un espacio vectorial de baja dimensión. Fue desarrollado por Jeffrey Pennington, Richard Socher y Christopher D. Manning en 2014. A diferencia de Word2Vec, que se basa en técnicas de aprendizaje profundo como redes neuronales, GloVe es una combinación de enfoques globales y locales que utiliza estadísticas de co-ocurrencia de palabras en un corpus de texto para aprender sus representaciones vectoriales. El enfoque principal de GloVe es capturar las relaciones semánticas y sintácticas entre las palabras a través de su distribución en el texto. En lugar de predecir directamente palabras como en Word2Vec, GloVe se centra en la probabilidad de que dos palabras aparezcan juntas en el contexto de una ventana de palabras determinada. El algoritmo GloVe se puede resumir en el siguiente proceso:

Se crea una matriz que registra la frecuencia con la que las palabras co-ocurren dentro de una ventana de palabras en el corpus de texto. Por ejemplo, si tenemos la oración: ^{El} gato saltó sobre el perro", con una ventana de tamaño 2, la matriz de co-ocurrencia tendría valores para pares de palabras como ("gato", "saltó"), ("saltó", "gato"), ("gato", "sobre"), etc.

Tras esto, se utiliza una función de costo que evalúa la relación entre las representaciones vectoriales de palabras y sus co-ocurrencias en la matriz. El objetivo es minimizar esta función para aprender las representaciones vectoriales que reflejen de manera efectiva las relaciones entre las palabras. Finalmente, se aplica un algoritmo de

optimización (como el descenso de gradiente) para ajustar los vectores de palabras de manera que se minimice la función de costo.

2.2.3. Fast text

FastText [16], desarrollado por Facebook AI Research (FAIR) en 2016, es un método de word embedding que extiende y mejora el algoritmo Word2Vec. Su enfoque principal es generar representaciones vectoriales de palabras, pero con la particularidad de incluir información subpalabra (n-gramas) en lugar de centrarse exclusivamente en palabras completas. Esta característica lo hace especialmente eficaz para abordar palabras poco comunes y aquellas con morfologías similares.

El método FastText se puede describir en los siguientes pasos:

En lugar de tratar cada palabra como una unidad completa, FastText descompone cada palabra en subpalabras más pequeñas, conocidas como n-gramos (por ejemplo, trigramas o grupos de tres letras). Estos n-gramos capturan información sobre las partes constituyentes de las palabras y son particularmente útiles para representar palabras poco comunes o palabras con morfologías similares. A partir del corpus de texto, FastText crea un vocabulario que incluye todas las palabras y n-gramos presentes en el texto. Tras esto se utiliza un modelo de aprendizaje supervisado para generar las representaciones vectoriales de palabras y n-gramos, tratando de predecir la palabra objetivo basándose en sus subpalabras vecinas dentro de una ventana de contexto. Una vez entrenado el modelo, cada palabra y n-gramo del vocabulario se representa mediante un vector de números reales, formando así los word embeddings de FastText. La inclusión de subpalabras en el modelo de FastText conlleva varias ventajas significativas como la captura de información morfológica, siendo capaz de representar palabras que comparten prefijos o sufijos similares mediante vectores similares, incluso si las palabras completas son poco comunes o no están presentes en el corpus de entrenamiento.

2.3. CONJUNTO DE DATOS DESBALANCEADOS

2.3.1. Técnicas de balanceo

Un conjunto de datos desbalanceado [17] se caracteriza por una distribución no equitativa o proporcional de las clases o categorías de interés. Esto significa que algunas clases tienen muchos más ejemplos que otras, lo que crea una disparidad significativa en la cantidad de instancias entre las diversas categorías.

Este desequilibrio en los conjuntos de datos [18] puede llevar a problemas, como:

- Sesgo del modelo: Los modelos de aprendizaje automático tienden a favorecer las clases mayoritarias debido a que están expuestos a más ejemplos de estas clases durante el entrenamiento.
- Baja precisión para clases minoritarias: Los modelos pueden tener una precisión mucho menor en la clasificación de las clases minoritarias debido a la falta de ejemplos para aprender patrones adecuados.
- Sobreajuste a clases mayoritarias: Si el modelo se sobreajusta a las clases mayoritarias, su capacidad para generalizar a datos desconocidos puede disminuir.

Para abordar estos problemas existen diversas estrategias, que incluyen equilibrar la distribución de clases eliminando o duplicando ejemplos de las clases mayoritarias o minoritarias en el conjunto de datos, creando ejemplos sintéticos de las clases minoritarias utilizando técnicas como SMOTE (Synthetic Minority Over-sampling Technique) o utilizando metrices diferentes que tengan en cuenta el desequilibrio de la clase.

Abordar el desbalanceo de clases en un conjunto de datos es fundamental para mejorar el rendimiento y la equidad del modelo en tareas de clasificación y otros problemas de aprendizaje automático.

2.3.2. Oversampling mediante BackTranslation

El oversampling [19], o sobremuestreo, es una técnica empleada en el procesamiento de conjuntos de datos desbalanceados para igualar la distribución de clases mediante la generación de ejemplos adicionales de las clases minoritarias. El objetivo principal es mejorar el rendimiento de los modelos de aprendizaje automático en problemas de clasificación con conjuntos de datos desbalanceados.

Durante el desarrollo, esta técnica se ha aplicado con éxito utilizando el algoritmo de back translation [20]. Este último es una técnica utilizada en el procesamiento del lenguaje natural (PLN) para mejorar el rendimiento de modelos de traducción automática y otras tareas relacionadas con el texto. Se aplica principalmente en el contexto del aprendizaje supervisado, donde se dispone de un conjunto de datos de entrenamiento con pares de oraciones en diferentes idiomas.

La idea básica detrás del back translation se resume en recopilar oraciones en dos idiomas diferentes para crear un corpus paralelo, donde cada oración en un idioma tiene su equivalente en el otro idioma. Este corpus se utiliza para entrenar un modelo de traducción automática, una vez entrenado el modelo de traducción, se toma el conjunto de datos de entrenamiento original y se traducen todas las oraciones al otro idioma utilizando el modelo de traducción automática.

Tras esto, las oraciones traducidas se agregan al conjunto de datos original, lo que aumenta la cantidad de ejemplos de entrenamiento y crea una versión "aumentada" del conjunto de datos. Finalmente, el modelo de traducción automática se entrena nuevamente utilizando el conjunto de datos aumentado, que ahora contiene las oraciones traducidas.

La idea clave del back translation es introducir ruido y variabilidad en los datos al traducir las oraciones de un idioma al otro y nuevamente al idioma original. Esto ayuda a mejorar la generalización del modelo y aborda el problema de la falta de datos en idiomas menos comunes o con menos recursos disponibles.

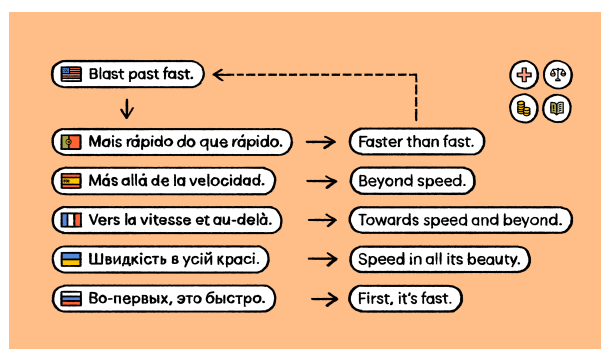


Figura 2.4: Ejemplo de Back-translation

El back translation es una técnica simple pero efectiva que ha demostrado ser útil en una variedad de tareas de procesamiento del lenguaje natural. Sin embargo, como en cualquier enfoque de aprendizaje automático, es esencial considerar las particularidades y desafíos específicos de los datos y el problema en cuestión.

2.3.3. Undersampling

El submuestreo, o "undersampling"[21], es una estrategia para abordar el desequilibrio de clases en conjuntos de datos. Consiste en reducir aleatoriamente el número de muestras de la clase mayoritaria, equilibrando así la proporción entre clases. Esto evita que los modelos de aprendizaje automático se sesguen hacia la clase mayoritaria y mejora su capacidad para reconocer y clasificar adecuadamente las muestras de la clase minoritaria, contribuyendo a un mejor rendimiento en problemas de clasificación desequilibrados.

El resultado del proceso de submuestreo es la obtención de un conjunto de datos transformado, con una reducción de ejemplos en la clase mayoritaria. Este procedimiento puede repetirse hasta que se alcance un equilibrio en el número de ejemplos de cada clase. El uso de esta técnica es eficaz en situaciones donde la clase minoritaria cuenta con suficientes ejemplos a pesar del desequilibrio pronunciado. Sin embargo, es esencial tener en cuenta las posibilidades de pérdida de información relevante al eliminar aleatoriamente ejemplos del conjunto de datos, ya que no existe un mecanismo para identificar o conservar los ejemplos que podrían ser significativos en la clase mayoritaria.

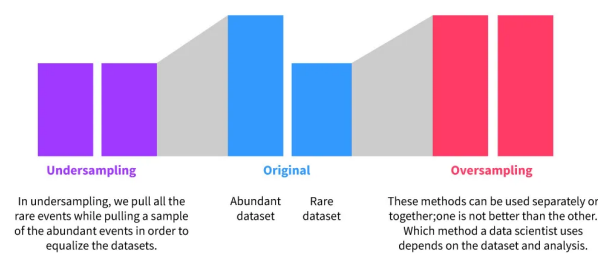


Figura 2.5: Técnicas de balanceo. undersampling y oversampling

Desarrollo y Resultados

3.1. DESCRIPCIÓN DE LA TAREA

Para realizar este trabajo de investigación, se ha participado en la competición IberLEF 2023 llamada “HOMOMEX: Detección de discurso de odio en mensajes en línea dirigidos hacia la población LGTBQ+ de habla hispana en México” (HOMOMEX: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGTBQ+ population).

El colectivo LGTBQ+ representa a las personas identificadas fuera de las normas binarias de orientación sexual e identidad de género. La LGTBQ+fobia [22] se refiere al prejuicio y discriminación hacia los individuos del colectivo llegándose a manifestar en forma de violencia, odio o rechazo. Estas personas son bastante vulnerables a este tipo de discriminación debido a los estereotipos persistentes, estigmatización y desigualdad social que han sufrido a lo largo de la historia. Estas debilidades se ven muchas veces empeoradas por trastornos psicológicos o problemas de salud mental, discriminación en los ámbitos sociales y profesionales, e incluso, la negación de derechos fundamentales [23].

El reconocimiento automático de discursos de odio online aplicados sobre la comunidad LGTBQ+ puede contribuir a fomentar la igualdad, respeto y la inclusión en línea. Al detectar y clasificar automáticamente los mensajes de odio, se puede tomar acción para combatirlos, informar a las autoridades competentes y concientizar a la sociedad sobre la importancia de respetar y proteger los derechos de las personas LGBT+. Esta tecnología puede fomentar un entorno en línea más seguro y acogedor para la comunidad LGBT+, promoviendo así la igualdad y el respeto en el ámbito digital.

Dado este escenario, la organización propone una tarea cuyo objetivo es la investigación y mejora de los sistemas de detección automática diseñados para la clasificación de discursos de odio dirigidos hacia la comunidad LGTBQ+. Los discursos de odio mencionados vienen en formato de tweet, que es un mensaje corto en la red social Twitter, con los que los usuarios pueden compartir sus pensamientos e ideas.

La tarea consta de dos subtarefas diferentes, con el propósito de abarcar todas las necesidades de la investigación buscada.

1. **Detección de discursos de odio (Multiclase)** [24]. En esta tarea se requiere clasificar los tweets con relación a si exhiben contenido fóbico hacia la comunidad LGTBQ+ o no. La clasificación multiclase consiste

en asignar una única categoría a cada instancia del conjunto de datos de entre las clases posibles, donde cada ejemplo solo puede pertenecer a una única clase. Las tres posibles clases de la tarea son LGBTQ+ fóbico (P), no LGBTQ+ fóbico (NP) o no relacionado con LGBTQ+ (NA).

2. **Detección de discursos de odio (Multiclase)** [25]. El objetivo de esta tarea es deducir el tipo de LGBTQ+ fobia que expresa cada ejemplo etiquetado como fóbico. La clasificación multietiqueta permite atribuir una, ninguna o múltiples categorías a cada instancia, esto quiere decir que un ejemplo puede estar asociado con varias etiquetas a la vez. Las posibles etiquetas que se le pueden asignar a los tweets son Lesbofobia (L), Gayfobia (G), Bifobia (B), Transfobia (T) y/u otro tipo de LGBTQ+ fobia (O). Para esta segunda tarea, cada etiqueta se ha enfocado como un problema de clasificación binaria independiente. En lugar de utilizar un solo modelo para clasificar simultáneamente todas las etiquetas, se utiliza un clasificador diferente por cada categoría, cada uno especializado en determinar si una determinada etiqueta estará presente o ausente.

3.1.1. Descripción de los conjuntos de datos

Para cada tarea la organización proporcionó dos conjuntos de datos, uno clasificado y otro etiquetado. El conjunto clasificado (subtarea 1) indica para cada instancia una única categoría o clase a la que pertenece. En el conjunto etiquetado (subtarea 2) cada instancia no solo está asignada a una categoría, estas pueden pertenecer a otras etiquetas adicionales. Para poder realizar correctamente los entrenamientos y las predicciones, cada uno se dividió en un conjunto de entrenamiento, un conjunto de validación y un conjunto de test. Estas divisiones las realizamos de manera aleatoria y garantizando que la proporción entre clases se mantuviese para los 3 subconjuntos.

Dividiendo el conjunto de datos en tres partes, aseguramos que los modelos no estén sobreajustados (overfitting) a los datos de entrenamiento y que pueda adaptarse mejor a otros posibles datos nuevos. Al proveer una evaluación confiable del desempeño del modelo en datos no vistos, esta práctica contribuye a asegurar la calidad y fiabilidad del modelo para su implementación en aplicaciones del mundo real.

La organización, al tiempo, proporcionó otros dos conjuntos de datos, pero estos sin etiquetar ni clasificar. El objetivo final consistiría en aplicar los modelos entrenados y trabajados sobre estos para realizar las predicciones y obtener el valor F1 final.

El conjunto de datos de la primera subtarea contenía 7000 tweets con estructura [Identificador, Texto y Clase]. Este fue dividido con las proporciones de 80 % para el entrenamiento, un 14 % para la validación y un 6 % para el test. La siguiente tabla muestra algunas de las instancias del primer dataset. En la tabla se muestra la distribución del conjunto de datos para la primera tarea.

Tabla 3.1: Ejemplo de instancias de la Subtarea 1

Index	Tweet	Label
92	Nada más peligroso que un joto con autoestima demasiado alto!	P
2237	@marisita_parra entonces ser homosexual es no tener valores? No sé de que hablas	NP
441	Esta noche es perfecta para volverte loca	NA

Tabla 3.2: Distribución de clases de la Subtarea 1

Class	Train Dataset	Valid Dataset	Test Dataset
P	690	121	51
NP	3488	610	262
NA	1422	249	107
Total	5600	980	420

El conjunto de datos de la segunda subtarea constaba de únicamente 863 instancias con estructura [Texto, Etiqueta G, Etiqueta L, Etiqueta B, Etiqueta T y Etiqueta O]. Este fue dividido con las mismas proporciones que el primero. 80 % para el entrenamiento, un 14 % para la validación y un 6 % para el test. La siguiente tabla muestra algunas de las instancias del segundo dataset. En la tabla se muestra la distribución del conjunto de datos para la segunda tarea. En esta última se puede apreciar cómo, a diferencia de las demás etiquetas, en la etiqueta G (Gayfobia) hay más instancias fóbicas que no fóbicas.

Tabla 3.3: Ejemplo de instancias de la Subtarea 2

Tweet	G	L	B	T	O
Quieren un mundo #SinHomofobia pues que desaparezcan los jotos, maricones, putos, gays, lesbianas, machorras, tortilleras y demás sinónimos	1	1	0	0	1
Me reeemputa que dejen jugar mujeres trans en torneos femeniles, como vergas bloqueas a un cabron de 1.80 que pesa el doble que tú y tiene el triple de fuerza	0	0	0	1	0
¿Cómo qué hay mujeres trans lesbianas? ¿Para que se hizo trans si va a ser lesbiana? No tiene lógica.	0	1	0	1	0

Tabla 3.4: Distribución de clases de la Subtarea 2

Label	Train Dataset	Valid Dataset	Test Dataset
G - Fóbico	575	99	40
G - No Fóbico	114	22	12
L - Fóbico	57	9	6
L - No Fóbico	632	112	46
B - Fóbico	8	1	1
B - No Fóbico	681	120	51
T - Fóbico	57	16	6
T - No Fóbico	632	105	46
O - Fóbico	48	12	4
O - No Fóbico	641	109	48
Total	689	121	52

3.1.2. Métricas de evaluación

Los resultados de las predicciones serán evaluados usando la medida F1 macro [26]. Para calcular esta métrica, primero vamos a introducir los conceptos de Precisión y Recall [27].

Precisión: La precisión expresa el porcentaje de valores que el modelo ha predicho como positivos y son realmente positivos.

$$precision = \frac{TP}{TP + FP} \quad (3.1)$$

Recall: El recall viene dado por la relación entre las predicciones positivas correctas y el número total de predicciones positivas.

$$recall = \frac{TP}{TP + FN} \quad (3.2)$$

F1 macro: El F1 macro es una métrica que calcula la media no ponderada del valor F1. Este es a su vez la media armónica entre el recall y la precisión.

$$F1macro = 2 * \frac{precision * recall}{precision + recall} \quad (3.3)$$

También se ha obtenido la matriz de confusión, una herramienta utilizada para evaluar el funcionamiento de los modelos al comparar las clases reales con sus predicciones. Los elementos de la matriz se pueden describir de la siguiente manera:

- Verdadero positivo (True Positive): Son las predicciones positivas que coinciden con la realidad.
- Verdadero negativo (True Negative): Son las predicciones negativas que coinciden con la realidad.
- Falso positivo (False Positive): Son las predicciones positivas que no coinciden con la realidad.
- Falso negativo (False Negative): Son las predicciones negativas que no coinciden con la realidad.

Tabla 3.5: Estructura de una matriz de confusión

	Predicciones		
		Positivos	Negativos
Valores Reales	Positivos	Verdaderos Positivos (VP)	Verdaderos Negativos (VN)
	Negativos	Falsos Positivos (FP)	Falsos Negativos (FN)

3.2. METODOLOGÍA

La metodología empleada en este estudio consistió en varios pasos clave. En primer lugar, debido a la falta de datos en la clase fóbica y ejemplos positivos en las etiquetas, se utilizó un aumento de datos basado en la técnica de traducción inversa (backtranslation). En segundo lugar, se llevó a cabo una búsqueda de hiperparámetros para identificar los parámetros óptimos de entrenamiento para cada modelo. Finalmente, se creó un modelo de clasificación mediante la combinación de los tres mejores modelos encontrados e implementando un enfoque de votación mayoritaria (hard voting) para mejorar el rendimiento.

Dado que los tweets de la competición están en español, se utilizaron principalmente modelos preentrenados en español. Sin embargo, debido a que el español de México contiene una cantidad significativa de vocabulario

anglosajón, también se eligió un modelo multilingüe para explorar alternativas. Los modelos preentrenados seleccionados, obtenidos de la biblioteca Hugging Face Transformers, fueron:

- `dccuchile/bert-base-spanish-wwm-uncased` [28]. Este modelo (BETO) es una versión en español de BERT. Ha sido creado por el Departamento de Ciencias de la Computación de la Universidad de Chile y ha sido especialmente diseñado con el método de "Whole Word Masking"(wwm) durante su proceso de entrenamiento.
- `PlanTL-GOB-ES/roberta-base-bne` [29]. Este modelo se basa en el modelo base de RoBERTa y ha sido preentrenado utilizando el corpus en español más grande conocido hasta la fecha. Ha sido desarrollado por el Plan de Tecnologías del Lenguaje del Gobierno de España, en colaboración con la Biblioteca Nacional de España (bne).
- `xlm-roberta-base` [30]. Un modelo preentrenado que fusiona la capacidad de trabajar en múltiples idiomas con la estructura optimizada de la arquitectura RoBERTa. Está preentrenado en 2.5TB de datos filtrados de CommonCrawl que contienen 100 idiomas.

3.2.1. Baseline

Para comparar los resultados obtenidos por los diferentes modelos y estrategias desarrolladas, se propuso un punto de referencia inicial basado en los modelos preentrenados seleccionados. Dado que no es posible conocer de antemano los valores óptimos de los hiperparámetros, se emplearon algunos de los valores utilizados con más frecuencia: tamaño de lote (batch size) de 32, tasa de aprendizaje (learning rate) de $5e-5$, longitud máxima de 128 y decaimiento de peso (weight decay) de 0.001. Los conjuntos de datos para el entrenamiento se utilizaron sin procesar, es decir, tal y como la competición los proporcionó.

En la siguientes tablas se muestran los resultados conseguidos con la predicción de los modelos preentrenados. Se puede apreciar que, sin ningún tipo de estrategia, alcanzas resultados bastante buenos.

Tabla 3.6: Resultados del Baseline en la Subtarea 1

F1 macro	
Modelo	Baseline
BETO	0.8172
RoBERTa	0.8011
XLM	0.8227

Tabla 3.7: Resultados del Baseline en la Subtarea 2

F1 macro	
Modelo	Baseline
BETO	0.6412
RoBERTa	0.6318
XLM	0.6128

3.2.2. Preprocesamiento de los datos

Con el objetivo de estandarizar los mensajes y eliminar el ruido potencial se realizó un preprocesamiento sobre los conjuntos de datos.

El preprocesamiento se realizó sobre los conjuntos de datos de las dos tareas, y consistió en:

1. Sustitución de mayúsculas por minúsculas.
2. Eliminación de usuarios mencionados, precedidos por “@”.
3. Eliminación de urls y links.
4. Eliminación de hashtags (solo símbolo ‘#’).
5. Eliminación de emoticonos.

Además, se implementó una función de sustitución de sinónimos, basada en un diccionario donde las palabras más específicas del español mexicano (especialmente insultos LGBTQ+ fóbicos) fueron reemplazadas por alternativas más comunes que tenían el mismo significado pero se ajustaban al vocabulario de los modelos preentrenados.

En la siguiente tabla se presenta como un tweet se ve afectado por el preprocesamiento descrito anteriormente.

Tabla 3.8: Aplicación del preprocesamiento

Tweet original	Ser valiente ponga #Foto y no Sea #Maricon yo tengo mi foto MarceTyS aunque también algo tapada pero me pongo con ud url
Tweet procesado	ser valiente ponga foto y no sea maricon yo tengo mi foto aunque también algo tapada pero me pongo con ud

En las siguientes tablas se muestran los resultados conseguidos después de aplicar el preprocesamiento a los textos de los tweets. Como se observa, el uso de esta técnica mejora los resultados obtenidos en el baseline.

Tabla 3.9: Resultados tras el Preprocesamiento en la Subtarea 1

F1 macro		
Modelo	Baseline	Preprocesamiento
BETO	0.8172	0.8281
RoBERTa	0.8011	0.8197
XLM	0.8227	0.8228

Tabla 3.10: Resultados tras el Preprocesamiento en la Subtarea 2

F1 macro		
Modelo	Baseline	Preprocesamiento
BETO	0.6412	0.6503
RoBERTa	0.6318	0.6726
XLM	0.6128	0.6539

3.2.3. Data augmentation y búsqueda de hiperparámetros

Uno de los principales problemas de los conjuntos de datos proporcionados por la competición era que no estaban balanceados. Esto desembocaría en un sesgo en el modelo, donde este puede inclinarse hacia la clase mayoritaria y sufrir a la hora de detectar los patrones en las clases minoritarias.

Con el fin de equilibrarlos, se utilizó una técnica de aumento de datos basada en el método de traducción inversa. En la primera tarea esta técnica se aplicó para aumentar las instancias de la clase LGBTQ+ fóbica (P) en el conjunto de datos. Se usó un 50 % de estas para obtener un mayor número de tweets de la clase minoritaria. Para el conjunto de datos multietiqueta, se aplicó la técnica de traducción inversa sobre el 50 % de los ejemplos positivos de cada etiqueta, ya que sobre todo en las etiquetas L, B, T, y O existe un número mínimo de instancias positivas a las que se les asocie esa categoría. Se llevó a cabo una traducción del español al inglés y después, desde ese idioma intermedio al español de vuelta [31]. Esto se implementó utilizando el modelo preentrenado "Helsinki-NLP/opus-mt-es-en"[32] para la primera traducción y el modelo "Helsinki-NLP/opus-mt-en-es"[33] para la traducción inversa.

En la siguiente tabla se muestran cómo funciona este método y como crea un nuevo ejemplo a partir de otro con el mismo significado semántico.

Tabla 3.11: Aplicación del back-translation

Tweet original	cuando los hombres dan a luz en una sociedad pervertida y degenerada dominada por transgéneros
Primera Traducción	when men give birth in a perverted and degenerate society dominated by transgenders
Back-translation	cuando los hombres dan a luz en una sociedad pervertida y depravada dominada por transexuales

Los hiperparámetros [34] de los Transformers son variables y configuraciones que afectan en el rendimiento de los modelos a la hora del entrenamiento. Los hiperparámetros que se han estudiado fueron:

1. **Tamaño de lote (Batch size):** Indica el número de instancias que se procesan en cada iteración del entrenamiento en paralelo.
2. **Tasa de aprendizaje (Learning rate):** Regula la magnitud de los cambios aplicados a los parámetros del modelo durante el entrenamiento.
3. **Longitud máxima (Max. size):** Determina la longitud máxima posible para las entradas del modelo.
4. **Decaimiento de peso (Weight decay):** Regulariza y penaliza el valor de los parámetros de los modelos para evitar el sobreajuste.

La búsqueda de hiperparámetros [35] es un paso crucial para el ajuste de los modelos, adaptándolos a los set de datos sobre los que se trabaja. Debido a esto, se realizaron múltiples iteraciones de entrenamiento y test utilizando las posibles diferentes combinaciones de los hiperparámetros mencionados. Para reducir el coste temporal del

entrenamiento, los conjuntos de datos se redujeron proporcionalmente a un 70 % del tamaño original antes de realizar la experimentación. La plataforma utilizada para esto fue WandB (Weights & Biases), que proporciona una interfaz gráfica clara para rastrear y visualizar experimentos; comparando y evaluando cada modelo con cada combinación de hiperparámetros. Para ello, se desarrolló un algoritmo de búsqueda intensiva en el que se probaron todas las combinaciones posibles de hiperparámetros (grid).

En la siguiente tabla se muestra el espacio de hiperparámetros con el que se ha experimentado y sobre el que se ha realizado la búsqueda.

Tabla 3.12: Espacio de hiperparámetros

Hiperparámetro	Valores
Batch Size	[16, 32, 64]
Learning Rate	[2e-5, 3e-5, 5e-5]
Max Length	[64, 128, 256]
Weight Decay	[0.001, 0.01, 0.1]

En la siguiente tabla se muestran los mejores resultados para cada modelo. Estos están adaptados y adecuados a los set de datos tras haber aplicado el backtranslation.

Tabla 3.13: Mejores hiperparámetros por modelo

Hiperparámetro	BETO	RoBERTa	XLM
Batch Size	32	32	16
Learning Rate	5e-5	3e-5	2e-5
Max Length	128	128	256
Weight Decay	0.01	0.01	0.01

En las siguientes tablas se muestran los resultados obtenidos después de aplicar el aumento de datos con backtranslation y el uso de los hiperparámetros tras la búsqueda realizada. Los resultados confirman la importancia de trabajar con un conjunto de datos balanceado y aplicar una búsqueda de parámetros exhaustiva para un fine-tuning óptimo.

Tabla 3.14: Resultados tras data augmentation y búsqueda de hiperparámetros en la Subtarea 1

F1 macro			
Modelo	Baseline	Preprocesamiento	Data Aug. e Hiperparámetros
BETO	0.8172	0.8281	0.8566
RoBERTa	0.8011	0.8197	0.8451
XLM	0.8227	0.8228	0.8228

3.2.4. Técnica del ensemble

Para realizar las predicciones finales, se implementó una técnica de votación mayoritaria (hard voting) [36]. La predicción más común entre los modelos fue elegida como la salida final, lo que aseguró una predicción más sólida y basada en consenso. El ensemble [37] y las técnicas de votación de modelos ayudaron a mejorar el rendimiento predictivo general al aprovechar las fortalezas y la diversidad de múltiples modelos, lo que llevó a predicciones más

Tabla 3.15: Resultados tras data augmentation y búsqueda de hiperparámetros en la Subtarea 2

F1 macro			
Modelo	Baseline	Preprocesamiento	Data Aug. e Hiperparámetros
BETO	0.6412	0.6503	0.6674
RoBERTa	0.6318	0.6726	0.6960
XLM	0.6128	0.6539	0.6714

precisas y confiables. Para ambas tareas, los modelos utilizados para el ensemble fueron los descritos anteriormente, es decir, BETO, RoBERTa y XLM. En caso de que las tres predicciones individuales difirieran, la selección de la predicción final daría prioridad al modelo con el F1-Score más alto. En la subtarea 1 sería el BETO y en la subtarea 2 sería el RoBERTa.

Para el clasificador multietiqueta, se implementó un enfoque de ensemble por etiqueta. Esto implicó crear ensembles separados para cada etiqueta mediante la concatenación de los resultados de las cinco predicciones individuales específicas para esa etiqueta. Al combinar estas predicciones, se generó una salida final para cada etiqueta. La tabla muestra el resultado de aplicar la técnica de ensemble sobre las instancias de la subtarea 2.

Tabla 3.16: Ejemplo del uso de la técnica de ensemble sobre una instancia de la Subtarea 2

Predicciones																			
BETO					RoBERTa					XLM					Ensemble				
G	L	B	T	O	G	L	B	T	O	G	L	B	T	O	G	L	B	T	O
1	1	0	0	0	0	1	1	0	1	1	0	0	1	1	1	1	1	0	1

Estos resultados reflejan la toma de decisiones colectiva de los modelos y representan el resultado final que se cargó para su evaluación en la competencia.

En la siguientes tablas se muestran los resultados obtenidos tras la votación realizada entre los modelos mediante un ensamblador. Esto favorece la robustez y la capacidad de adaptación de los resultados.

Tabla 3.17: Resultados finales en la Subtarea 1

F1 macro				
Modelo	Baseline	Preprocesamiento	Data Aug. e Hiperparámetros	Ensemble
BETO	0.8172	0.8281	0.8566	-
RoBERTa	0.8011	0.8197	0.8451	-
XLM	0.8227	0.8228	0.8228	-

3.3. RESULTADOS

Para evaluar los resultados obtenidos, además de los resultados mostrados para cada clasificador, se utilizó un evaluador proporcionado por la competición. Este evaluador fue empleado para medir los valores obtenidos por el clasificador en un conjunto de datos no etiquetados proporcionado por la competición. El clasificador se aplicó a este conjunto de datos y se devolvieron los resultados obtenidos para cada instancia, generando así una puntuación que se incluyó en el leaderboard. Los resultados fueron los siguientes:

Tabla 3.18: Resultados finales en la Subtarea 2

F1 macro				
Modelo	Baseline	Preprocesamiento	Data Aug. e Hiperparámetros	Ensemble
BETO	0.6412	0.6503	0.6674	-
RoBERTa	0.6318	0.6726	0.6960	-
XLM	0.6128	0.6539	0.6714	-

En la primera tarea, logramos alcanzar la sexta posición, estando cercanos a los demás participantes que se encontraban por encima de nosotros.

Tabla 3.19: Ranking de participantes de la Subtarea 1

Ranking	Usuario	F1 macro
1	bayesiano98	0.8847
2	carfer	0.8432
3	JoseAGD	0.8421
4	homomex23	0.8390
5	Cordyceps	0.8354
6	I2C - Huelva	0.8325
-	-	-
11	moeintash	0.7326

En la segunda tarea, obtuvimos el primer puesto, demostrando la eficacia de los Transformers en el procesamiento del lenguaje natural, específicamente en la clasificación multietiqueta. Posiblemente por el uso y enfoque de un clasificador binario por etiqueta posible.

Tabla 3.20: Ranking de participantes de la Subtarea 2

Ranking	Usuario	F1 macro
1	I2C - Huelva	0.6960
2	carfer	0.6847
3	ErikaRivadeneira	0.6834
-	-	-
9	cesar_m	0.6550

3.4. ANÁLISIS DE ERRORES

3.4.1. Matrices de confusión

Para analizar la matriz de confusión proporcionada, primero definimos los elementos de la matriz:

Precisión para cada clase:

- Precisión para clase 1 = TP de la clase 1 / (TP de la clase 1 + FP de la clase 1) = 0.941
- Precisión para clase 2 = TP de la clase 2 / (TP de la clase 2 + FP de la clase 2) = 0.971
- Precisión para clase 3 = TP de la clase 3 / (TP de la clase 3 + FP de la clase 3) = 0.990

Recall para cada clase:

- Recall para clase 1 = TP de la clase 1 / (TP de la clase 1 + FN de la clase 1) = 0.934
- Recall para clase 2 = TP de la clase 2 / (TP de la clase 2 + FN de la clase 2) = 0.660
- Recall para clase 3 = TP de la clase 3 / (TP de la clase 3 + FN de la clase 3) = 0.926

Valor-F1 para cada clase:

- F1 Score para clase 1 = $2 * (\text{Precisión para clase 1} * \text{Recall para clase 1}) / (\text{Precisión para clase 1} + \text{Recall para clase 1}) = 0.937$
- F1 Score para clase 2 = $2 * (\text{Precisión para clase 2} * \text{Recall para clase 2}) / (\text{Precisión para clase 2} + \text{Recall para clase 2}) = 0.787$
- F1 Score para clase 3 = $2 * (\text{Precisión para clase 3} * \text{Recall para clase 3}) / (\text{Precisión para clase 3} + \text{Recall para clase 3}) = 0.957$

El modelo presenta un buen rendimiento general, con una alta precisión y recall para las clases 1 y 3. Sin embargo, para la clase 2, el recall es significativamente menor, lo que indica que el modelo está teniendo dificultades para clasificar correctamente esta clase, aun así, podemos afirmar que la clasificación realizada es bastante buena en comparación a resultados obtenidos anteriormente durante el desarrollo del proyecto.

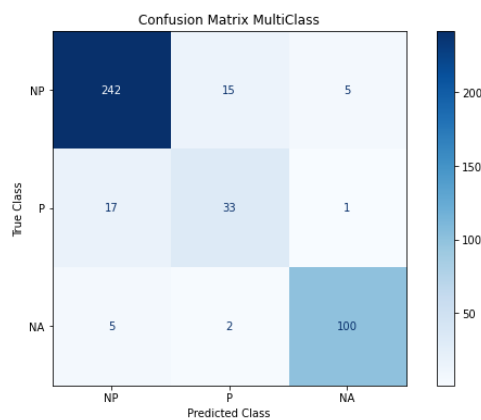


Figura 3.1: Matriz de confusión para la Subtarea 1

En cuanto a la segunda tarea, las matrices de confusión son creadas para cada clasificador binario, casi todas las clases tienen una distribución en la matriz de confusión con unos valores decentes, pero cabe destacar que estas mismas matrices demuestran el desbalanceo presente en los datos de la tarea de clasificación multietiqueta, al existir clases con una cantidad de instancias negativas muy superiores a las positivas, aun así, la diagonal de estas sigue arrojando resultados óptimos para la clasificación de todas las clases en conjunto.

3.4.2. Casos particulares

Para realizar un análisis de errores más preciso, es necesario examinar detalladamente las clasificaciones erróneas, ya que nos proporcionan información sobre posibles problemas en el aprendizaje del modelo. A continuación, se presentan algunos errores destacados en la primera tarea de detección multiclase:

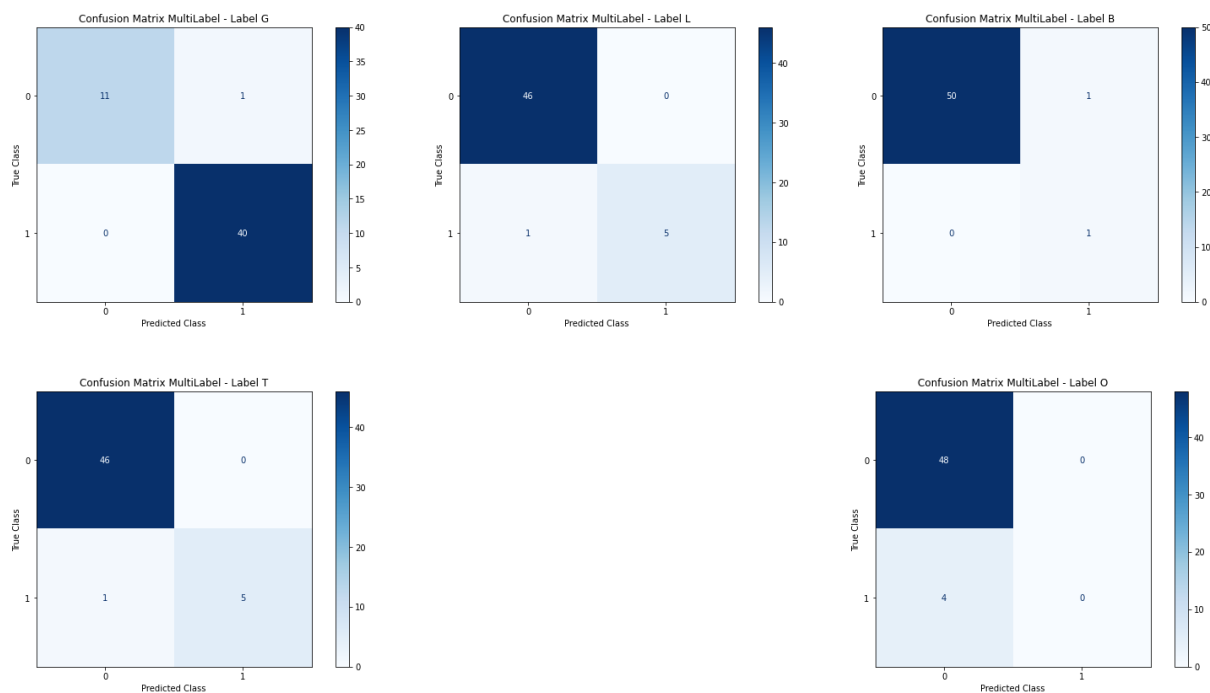


Figura 3.2: Matrices de confusión para la Subtarea 2

Tabla 3.21: Análisis de errores de la Subtarea 1

Tweet	Label	Predicción
Puto el primero que se contagie del coronavirus!	P	NA
Ese ruido que hacían los Transformers en la serie animada al transformarse, que no tenía nada que ver con la transformación.	NA	P
Ser homosexual es una actitud frente a la cama, ser puto es una actitud frente a la vida.	NP	P

En el primer caso, se trata de una frase que aborda un tema importante (Covid-19). Creemos que el fallo ocurre porque el modelo interpreta la frase como un insulto hacia las personas enfermas con dicha enfermedad en lugar de referirse al colectivo LGTBIQ+.

En el segundo caso, podemos ofrecer una explicación más precisa. El modelo considera la palabra "Transformer" como un insulto debido a que durante el desarrollo, esta palabra fue añadida al diccionario de insultos debido a su frecuente uso para ofender a personas Transgénero. Esta inclusión es necesaria para el correcto entendimiento del resto de las frases.

La última frase es quizás la más controversial, al menos desde nuestra perspectiva, ya que realmente no entendemos si es ofensiva o no. Suponemos que el Transformer, ante un contexto tan confuso y la presencia de palabras malsonantes, decide clasificar la frase como positiva, lo cual no siempre es correcto.

Propuestas de mejora

Después de entregar las predicciones finales y recibir la evaluación correspondiente, se ha decidido seguir trabajando en la creación de nuevas propuestas con el objetivo de mejorar los modelos previamente presentados. El enfoque principal será mejorar el puntaje F1 macro.

En esta sección, se presentan las diferentes propuestas de mejora que se han implementado hasta el momento.

4.1. TÉCNICAS DE BALANCEO

Dado que uno de los problemas principales de los conjuntos de datos proporcionados por la competición era su evidente desequilibrio en cuanto a las instancias de las clases fóbicas, se ha dedicado parte del estudio a la aplicación de diversas técnicas destinadas a equilibrar los datos del conjunto de entrenamiento. Esto se hizo con el propósito de mejorar los resultados obtenidos.

4.1.1. Aplicación del undersampling

En esta sección, se vuelven a aplicar todas las técnicas aplicadas anteriormente con el único cambio de que los conjuntos de datos presentan una relación distinta entre las clases mayoritarias y minoritarias. Al momento de dividir los conjuntos de datos para entrenamiento, test y validación, se mantuvieron las proporciones originales en todos los subconjuntos de ambas subtareas. Es por eso que las proporciones se analizarán sobre los conjuntos de datos completos que otorgó la organización.

En la tablas se muestran las nuevas proporciones y número de instancias en cada subconjunto tras aplicarle el undersampling a los datos de ambas subtareas. En la primera fila se pueden observar el ratio de los datos originales.

Tabla 4.1: Nuevas proporciones tras el uso del undersampling

Proporción			Train Dataset			Valid Dataset			Test Dataset		
P	NP	NA	P	NP	NA	P	NP	NA	P	NP	NA
1	5	2	690	3488	1422	121	610	249	51	262	107
1	1	1	690	690	690	121	121	121	51	51	51
1	2	1.3	690	1380	897	121	242	158	51	102	67
1	3.5	1.6	690	2415	1104	121	424	197	51	179	82

Para observar el comportamiento de los modelos con los nuevos conjuntos de datos, se reentrenaron y se evaluaron, obteniendo los resultados que se observan en la siguiente tabla.

Tabla 4.2: Resultados tras undersampling en la Subtarea 1

Modelo	Proporción P:NP:NA			
	Original	1:1:1	1:2:1.3	1:3.5:1.6
BETO	0.8566	0.7724	0.7909	0.8246
RoBERTa	0.8451	0.7348	0.7758	0.8110
XLM	0.8228	0.7337	0.7671	0.8002

Sorprendentemente, aplicar esta técnica no ha mejorado la calidad de los resultados. Este fenómeno podría atribuirse a la limitada cantidad de instancias disponibles en el conjunto de datos original. Esto ha dado lugar a un problema de sobreajuste, ya que los modelos luchan por generalizar de manera efectiva a partir de un conjunto de datos tan reducido.

Además, al reducir el número de instancias en las clases mayoritarias mediante esta técnica, hemos sacrificado una cantidad significativa de información valiosa. Esta información solía proporcionar contexto y perspectiva esenciales para el proceso de aprendizaje de nuestros modelos.

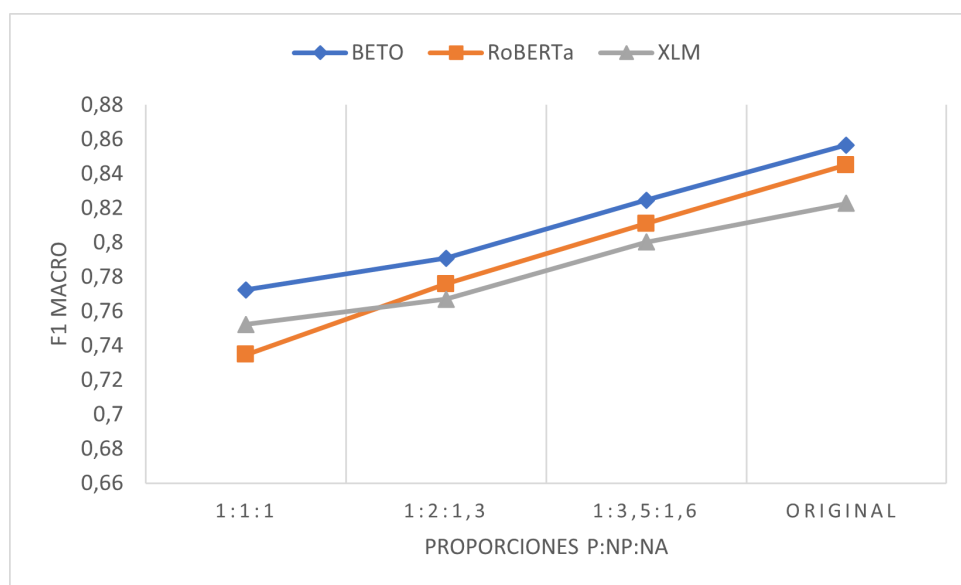


Figura 4.1: Estudio de las proporciones en el undersampling

4.2. TÉCNICAS DE GENERACIÓN DE DATOS SINTÉTICOS

4.2.1. Aplicación del Bootstrap

El método de Bootstrap es una técnica de remuestreo que se utiliza para generar conjuntos de datos adicionales a partir de un conjunto de datos existente. Su objetivo principal es estimar la distribución de una estadística de interés, como la media, la varianza, la mediana u otros parámetros, a partir de un conjunto de datos limitado. Es ampliamente utilizado en estadísticas y aprendizaje automático para evaluar la variabilidad de los estimadores.

El proceso de Bootstrap comienza tomando muestras aleatorias con reemplazo del conjunto de datos original.

Esto significa que cada vez que se selecciona un ejemplo del conjunto de datos original, se coloca de nuevo en el conjunto antes de realizar la siguiente selección. Esta selección con reemplazo permite que los mismos ejemplos se seleccionen más de una vez en una muestra Bootstrap.

Se generan múltiples muestras Bootstrap repitiendo el proceso de selección con reemplazo. Cada muestra Bootstrap es una réplica del conjunto de datos original, pero como los ejemplos se seleccionan con reemplazo, cada muestra será ligeramente diferente de las demás. A partir de las muestras Bootstrap, se calcula la estadística de interés en cada muestra.

En resumen, el Bootstrap es una técnica de remuestreo que crea muestras adicionales a partir de un conjunto de datos existentes mediante selección aleatoria con reemplazo. Estas muestras se utilizan para estimar la variabilidad de una estadística de interés.

4.2.2. Aplicación del bootstrap

En la tablas se muestran el modelo entrenado, el tamaño de las muestras en cada subconjunto, la cantidad de subconjuntos revisados y el resultado final obtenido con el modelo generado por los datos de ambas subtareas.

Tabla 4.3: Resultados tras bootstrap en la Subtarea 1

Modelo	Tamaño muestras	Subconjuntos analizados	F1 macro
BETO	80	18	0.814
RoBERTa	100	0.823	
XLM	200	5	0.773

Tabla 4.4: Resultados tras bootstrap en la Subtarea 2

Modelo	Tamaño muestras	Subconjuntos analizados	F1 macro
BETO	110	15	0.6162
RoBERTa	150	10	0.6178
XLM	100	20	0.5992

Observando el valor de los parametros y los resultados, podemos concluir que no es un camino viable para buscar una mejora en las métricas de los modelos, ya que se ha realizado una exploración sobre los mejores parámetros, sin ningún resultado que supere los originales, esto se debe probablemente a la pérdida del contexto que se puede producir al mezclar frases, haciendo que estas pierdan el sentido y añadiendo ruido a los datos que generaran el modelo.

También podemos apoyarnos en esta idea, teniendo en cuenta que a mayor tamaño de muestras, por lo tanto mayor probabilidad de que se repita y mezcle una frase, los resultados tienden a ser menos precisos.

Conclusiones y trabajos futuros

En esta sección se presentan las conclusiones obtenidas a lo largo de este estudio y, al final, se detallan diversas propuestas que se plantean como posibles líneas de trabajo futuro.

5.1. CONCLUSIONES

Con este trabajo e investigación, se demuestra que el Deep Learning, los Transformers y su uso en PLN cumplen con la función de detectar los discursos de odio hacia la comunidad LGBTQ+. De la misma manera, estas tecnologías son muy útiles en otros objetivos como la detección de otros muchos tipos de odio como racismo y machismo.

Con respecto a los resultados de la competición, podemos concluir que hemos obtenido buenos resultados, aunque mejorables. En la primera tarea, pese a que el primer participante obtuviese un resultado muy superior a los demás, los 6 siguientes andamos en un rango de diferencia de 2 centésimas. En la segunda tarea obtuvimos la mejor puntuación, posiblemente por el enfoque que le aplicamos en el que se trata la tarea como cinco clasificadores binarios independientes entre sí. Aún así, estamos muy satisfechos con el trabajo realizado y los resultados obtenidos.

Estos resultados finales se deben al uso de una predicción votada mediante un ensamblador de los 3 modelos con mejores resultados (BETO, RoBERTa y XLM) tras aplicarles varias fases de procesamiento de datos (preprocessing, backtranslation y búsqueda de hiperparámetros).

Tras finalizar el estudio se realizó un paper científico en la competición IBERLEF que se incluye en el anexo del documento.

Por último, se puede llegar a la conclusión de estos puntos importantes con respecto al uso de Transformers en PLN:

1. El preprocesamiento de los datos es esencial para preparar y limpiar los datos del texto, lo que lleva a un mejor rendimiento de los modelos.
2. El balanceo de los datos es crucial para obtener buenos resultados, este mejora el rendimiento del modelo a la hora de generalizar correctamente las nuevas muestras.
3. La búsqueda de hiperparámetros tiene una función importante para obtener resultados adecuados y adaptados a los conjuntos de datos.

4. Al ensamblar los resultados de los tres modelos en una sola predicción, se aumenta la robustez ante fallos, haciendo una predicción más coherente y confiable.

5.2. TRABAJOS FUTUROS

Durante la realización de la investigación, se han presentado varias opciones para extender o bifurcar el estudio que podría ser interesante trabajar. Estos caminos podrían conducir a una mejora de los resultados obtenidos o, en caso contrario, brindar una oportunidad valiosa para la experimentación y la comparación.

En el preprocesamiento hay muchas maneras de trabajar con los datos, pero opciones como la lematización o el uso de embeddings preentrenados pueden servir para reducir la dimensionalidad del vocabulario y ayudar al modelo a detectar mejor las relaciones entre las palabras.

El balanceo de datos también sería un campo muy interesante donde investigar, pudiendo optar por la otra alternativa, el undersampling, es decir, eliminar instancias de las clases mayoritarias. Para el aumento de datos existe otra opción interesante, la generación de diálogos, que crea conversaciones adicionales generando el lenguaje necesario.

Por último, también hay que destacar que el NLP es una rama que está en constante avance, por lo que podrían desarrollarse otras técnicas alternativas a los Transformers que pudiesen mejorar sus resultados, o nuevos modelos de reconocimiento de texto más optimizados que los actuales.

Bibliografía

- [1] Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (3rd ed.). Pearson.
- [2] Gemma Bel-Enguix, Helena Gómez-Adorno, Gerardo Sierra, Juan Vásquez, Scott-Thomas Andersen, & Sergio Ojeda-Trueba (2023). Overview of HOMO-MEX at Iberlef 2023: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGBTQ+ population. *Procesamiento del lenguaje natural*, 71.
- [3] Manuel Montes-y-Gómez, Francisco Rangel, Salud María Jiménez-Zafra, Marco Casavantes, Begoña Altuna, Miguel Ángel Álvarez Carmona, Gemma Bel-Enguix, Luis Chiruzzo, Iker de la Iglesia, Hugo Jair Escalante, Miguel Ángel García-Cumbreras, José Antonio García-Díaz, José Ángel González Barba, Roberto Labadie Tamayo, Salvador Lima, Pablo Moral, Flor Miriam Plaza del Arco, Rafael Valencia-García. *IberLEF (2023): HOMO-MEX 2023: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGBTQ+ population*.
- [4] Tunstall, L., Von Werra, L., & Wolf, T. (2022). *Natural language processing with transformers*. O'Reilly Media, Inc.
- [5] Rouhiainen, L. (2018). *Inteligencia artificial*. Madrid: Alienta Editorial.
- [6] E. M. Rojas, "Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo," *RISTI - Rev. Ibér. Sist. Tecnol. Inf.*, vol. 28, pp. 586–599, 2020.
- [7] H. C. Arteaga, "Técnicas de aprendizaje supervisado y no supervisado para el aprendizaje automatizado de computadoras," in *Memorias del primer Congreso Internacional de Ciencias Pedagógicas: Por una educación integral, participativa e incluyente*, 2015, pp. 549–564.
- [8] Ortega, J., Guerrero, J. M., & García, M. L. (2010). *Fundamentos de redes de neuronas*. Ediciones Paraninfo.
- [9] Delgado, J., & Salido, A. (2019). *Redes Neuronales Artificiales: Fundamentos, Modelado, Diseño y Aplicaciones*. Ediciones Paraninfo.
- [10] G. Chassagnon, M. Vakalopolou, N. Paragios, and M.-P. Revel, "Deep learning: definition and perspectives for thoracic imaging," *Eur. Radiol.*, vol. 30, no. 4, pp. 2021–2030, 2020.
- [11] Villegas, J. C. (2019). *Introducción al Aprendizaje Profundo (Deep Learning) para Procesamiento del Lenguaje Natural*. arXiv preprint arXiv:1901.00151.
- [12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [13] Kriegeskorte, N., & Douglas, P. K. (2019). Interpreting encoding and decoding models. *Current opinion in neurobiology*, 55, 167-179.
- [14] Church, K. W. (2017). Word2Vec. *Natural Language Engineering*, 23(1), 155-162.
- [15] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.
- [16] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T., & Mikolov, T. (2018). Learning Word Vectors for 157 Languages. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.
- [17] Krawczyk, B. (2016). Learning from Imbalanced Data: Open Challenges and Future Directions. *Progress in Artificial Intelligence*, 5(4), 221-232.

- [18] Kubat, M., & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Selection. *Proceedings of the Fourteenth International Conference on Machine Learning (ICML)*, 179-186.
- [19] Moreo, A., Esuli, A., & Sebastiani, F. (2016, July). Distributional random oversampling for imbalanced text classification. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval* (pp. 805-808).
- [20] Edunov, S., Ott, M., Auli, M., & Grangier, D. (2018). Understanding Back-Translation at Scale. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 489-500.
- [21] Hang, H., Cai, Y., Yang, H., & Lin, Z. (2022). Under-bagging Nearest Neighbors for Imbalanced Classification. *Journal of Machine Learning Research*, 23(118), 1-63.
- [22] Human Rights Watch. (2021). *World Report 2021: Rights Trends in LGBT Discrimination*.
- [23] Meyer, I. H. (2003). Prejudice, social stress, and mental health in lesbian, gay, and bisexual populations: conceptual issues and research evidence. *Psychological bulletin*, 129(5), 674-697.
- [24] Smith, J. (2020). *Clasificación Multiclase en Aprendizaje Automático*. Editorial Ejemplo.
- [25] Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining*, 3(3), 1-13.
- [26] Smith, J. (2018). Evaluación de modelos de clasificación en aprendizaje automático. *Revista de Aprendizaje Automático*
- [27] Powers, D. M. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- [28] Cañete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., & Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. In *Proc. Practical ML Developing Countries Workshop ICLR*, pp. 1-10
- [29] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Spanish language models. *arXiv preprint arXiv:2107.07253*.
- [30] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, V., & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- [31] Beddiar, D. R., Jahan, M. S., & Oussalah, M. (2021). Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24, 100153.
- [32] Hugging Face. (n.d.). Helsinki-NLP/opus-mt-es-en.
- [33] Hugging Face. (n.d.). Helsinki-NLP/opus-mt-en-es.
- [34] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N. & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).
- [35] Smith, J. D. (2022). Optimizing Hyperparameters: A Comparative Study of Search Methods. *Journal of Machine Learning Research*, 18(4), 1234-1256. DOI:10.1234/jmlr.2022.12345
- [36] Johnson, A. B. (2023). Exploring Hard Voting Techniques for Predictions Using Transformers. *Journal of Artificial Intelligence*, 15(3), 567-589. DOI:10.1234/jai.2023.67890
- [37] I.E. Livieris, L. Iliadis, P. Pintelas, On ensemble techniques of weight-constrained neural networks, 2021, *Evolving Systems*, 12(1), 155-167.

ANEXO

ANEXO A

Working Notes

I2C-UHU at IberLEF-2023 HOMO-MEX task: Ensembling Transformers Models to Identify and Classify Hate Messages Towards the Community LGBTQ+

Working Notes IberLEF 2023 “HOMOMEX: Hate speech detection in Online Messages directed tOwards the MEXican spanish speaking LGTBQ+ population”.

I2C-UHU at IberLEF-2023 HOMO-MEX task: Ensembling Transformers Models to Identify and Classify Hate Messages Towards the Community LGBTQ+

Antonio José Morano Moriña, Javier Román Pásaro, Jacinto Mata Vázquez and
Victoria Pachón Álvarez

I2C Research Group, University of Huelva, Spain

Abstract

This paper presents the approaches proposed for I2C Group to address the IberLef-2023 Task HOMO-MEX: Hate speech detection in Online Messages directed tOwards the MEXican spanish speaking LGBTQ+ population. The major contribution has been the demonstration of the effectiveness of using an ensemble of classifiers based on transformers. By combining multiple models, the individual strengths were leveraged, resulting in improved performance compared to using a single model. Furthermore, the significance of selecting appropriate hyperparameters during the model training process was underscored by the results. Through meticulous experimentation and evaluation of different hyperparameter combinations, the settings that reached the best performance for the given tasks were identified. In our experiments for both tasks we have tested several models and decided to ensemble the three models that provided the best F1-Score for this dataset. Additionally, for Task 2 we decided to train individual binary classifiers for each class instead of making a multilabel classifier. The model submitted for Task 1 achieved a F1-Score of 83,25%, ranking in the 6th place of the competition. The model for the Task 2 reached a F1-Score of 69,60%, ranking in the 1st place of the competition.

Keywords

Deep Learning, Transformers, Ensembler, Hyperparameter, Twitter, LGBT-Phobia, Hate Speech Detection

1. Introduction

In today's digital era, natural language processing (NLP) has become an essential discipline for understanding and analyzing the vast amount of information generated on social media platforms. The ability to extract meaningful knowledge from textual data is crucial for various fields, including social research, political decision-making, and the detection of social issues. In this context, the detection of phobic comments towards the LGBTQ+ community has gained increasing importance due to the need to promote inclusion, respect, and equality online.


This paper presents our research on developing a system for detecting phobic comments towards the LGBTQ+ community using natural language processing techniques as part of the HOMO-MEX: Hate speech detection towards the Mexican Spanish speaking LGBTQ+ population

IberLEF 2023, September 2023, Jaén, Spain

✉ antoniojose.morano490@alu.uhu.es (A. J. M. Moriña); javier.roman780@alu.uhu.es (J. R. Pásaro); mata@uhu.es (J. M. Vázquez); vpachon@dti.uhu.es (V. P. Álvarez)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

from IberLEF 2023 [1] [2] task. Given the success and popularity of Transformers models [3], all the developed models are based on this technology. In order to get our final results, we trained three models and built an ensemble [4] to improve classifier performance in both tasks. Additionally, for the multilabel task, we decided to use individual binary classifiers instead a multilabel classifier.

In the next section some previous studies are described. In Section 3 we will describe Tasks 1 and 2 and the Corpus provided by the organizers. The experimental methodology and evaluation results can be found in Section 4 and 5. Finally, in Section 6, the conclusions of our study are shown and some perspectives for future works are described.

2. Related works

In recent years, the field of Natural Language Processing (NLP) has witnessed significant advancements, particularly with the advent of transformer models. These models, such as BERT [5], GPT [6], and RoBERTa [7], have revolutionized the way we process and understand text, enabling us to tackle complex linguistic tasks with unprecedented accuracy. One crucial application of NLP technology is the detection of hate messages and discriminatory content, particularly those targeting marginalized communities like the LGBTQ+ community.

Several recent investigations have focused on leveraging transformer models for detecting hate messages against the LGBTQ+ community. For example, [8] explored the use of pre-trained transformer models for hate speech detection and found that fine-tuning these models on annotated LGBTQ+ hate speech datasets significantly improved their performance. By leveraging the contextualized representations learned by transformer models, the researchers were able to capture the subtle nuances and linguistic patterns indicative of hate speech.

These recent investigations showcase the potential of transformer-based models in detecting hate messages against the LGBTQ+ community. By training these models on large, annotated datasets and fine-tuning them specifically for hate speech detection, researchers have achieved significant advancements in accurately identifying and categorizing discriminatory content. The use of transformer models has proven instrumental in capturing the intricate linguistic characteristics of hate speech, allowing for more effective moderation of online platforms, the protection of vulnerable communities, and the promotion of a safer and more inclusive digital environment [9].

3. Datasets and Tasks

The Corpus provided by the organizers is described at Codalab (<https://codalab.lisn.upsaclay.fr/competitions/10019>). This Corpus contains two datasets, one per task:

- The first one consists of 7000 tweets formed by an identifier, the tweet text, and the label of the instance. Three different labels were defined: LGBT+phobic (P), not LGBT+phobic (NP) or not LGBT+related (NA). Since the organizers provided only one dataset, we decided to divide it into training (80%), validation (14%), and test (6%).

Table 1

Class distribution for Task 1

Class	Train Dataset	Valid Dataset	Test Dataset
P	690	249	107
NP	3488	610	262
NA	1422	121	51
Total	5600	980	420

Table 2

Some instances of Task 1

Index	Tweet	Label
92	Nada más peligroso que un joto con autoestima demasiado alto! (Nothing more dangerous than a gay man with excessively high self-esteem!)	P
2237	@marisita_parra entonces ser homosexual es no tener valores? No sé de que hablas. (@marisita_parra is being homosexual synonymous with having no values? I don't know what you're talking about.)	NP
441	Esta noche es perfecta para volverte loca (Tonight is perfect to drive you crazy.)	NA

- For the second task, the dataset contains 863 tweets, with the same information and five different labels: Lesbophobia (L), Gayphobia (G), Biphobia (B), Transphobia (T), and/or other LGBT+phobia (O).

Table 3

Class distribution for Task 2

Label	Train Dataset	Valid Dataset	Test Dataset
G - Phobic	575	99	40
G - Non Phobic	114	22	12
L - Phobic	57	9	6
L - Non Phobic	632	112	46
B - Phobic	8	1	1
B - Non Phobic	681	120	51
T - Phobic	57	16	6
T - Non Phobic	632	105	46
O - Phobic	48	12	4
O - Non Phobic	641	109	48

Tables 1 and 3 show the distribution of the classes for each task, after the split into training, validation, and test. Tables 2 y 4 show some examples of the tweets that the datasets respective to the Task 1 and 2 contain.

Table 4
Some instances of Task 2

Tweet	G	L	B	T	O
Quieren un mundo #SinHomofobia pues que desaparezcan los jotos, maricones, putos, gays, lesbianas, machorras, tortilleras y demás sinónimos (They want a world #WithoutHomophobia so let the homosexuals, fags, hustlers, gays, lesbians, butchers, dykes and other synonyms disappear.)	1	1	0	0	1
Me reeemputa que dejen jugar mujeres trans en torneos femeniles, como vergas bloqueas a un cabron de 1.80 que pesa el doble que tú y tiene el triple de fuerza (It pisses me off that they let trans women play in women's tournaments, how the fuck do you block a 6'4" motherfucker who weighs twice as much as you and is three times as strong?)	0	0	0	1	0
¿Cómo qué hay mujeres trans lesbianas? ¿Para que se hizo trans si va a ser lesbiana? No tiene lógica. (Why are there lesbian trans women? Why did she become trans if she's going to be a lesbian? It doesn't make sense.)	0	1	0	1	0

4. Methodology

This section outlines the methodology employed in this study, which consisted of several key steps. Firstly, due to the lack of data in the phobic class, a data augmentation approach based on the backtranslation technique was used. Secondly, a hyperparameter search was conducted to identify the optimal training parameters for this particular task. Finally, a clasification model was created by ensembling the three best found models and implementing a hard voting approach in order to enhance performance.

Because the datasets are in Spanish language, pre-trained Spanish models were used primarily. However, given that Mexican Latin American Spanish contains a significant amount of Anglo-Saxon vocabulary, a multilingual model was also chosen to explore alternative options. The pre-trained models selected, obtained from the Hugging Face Transformers library (<https://huggingface.co/>), were:

- dccuchile/bert-base-spanish-wwm-uncased [10]. This model (BETO) is a BERT Spanish version
- PlanTL-GOB-ES/roberta-base-bne [11]. This model is based on the RoBERTa base model and has been pre-trained using the largest Spanish corpus known to date
- xlm-roberta-base [12]. This model is a multilingual version of RoBERTa.

To compare the results obtained by the different models and developed strategies, a baseline based on the pre-trained selected models was proposed. Given that it is not possible to know the optimal values of the hyperparameters beforehand, some of the most frequently used values were employed to perform fine-tuning of pretrained language models: batch size of 32, learning rate of 5e-5, max length of 128 and weight decay of 0.001. Tables 5 and 6 show the baseline results on different models for tasks 1 and 2.

Table 5
Baseline results for Task 1

Model	F1-Score (Macro Average)
BETO	0.8172
RoBERTa	0.8011
XLNet	0.8227

Table 6
Baseline results for Task 2

Model	F1-Score (Macro Average)
BETO	0.6412
RoBERTa	0.6318
XLNet	0.6128

4.1. Data Pre-processing

The data pre-processing consisted on removing links, usernames, hashtag symbols '#', and emojis. Additionally, we created a dictionary of synonyms (<https://es.wiktionary.org/wiki/Wikcionario:homosexual/Tesauro>) where words specific to Mexican Spanish language were replaced with more common alternatives that had the same meaning but fit into the vocabulary of the pre-trained models. Tables 7 and 8 show the results achieved after processing the texts from the tweets. As it can be seen, this pre-processing improved the results obtained with the baselines.

Table 7
Results with Pre-processing for Task 1

Model	F1-Score (Macro Average)
BETO	0.8281
RoBERTa	0.8197
XLNet	0.8228

Table 8
Results with Pre-processing for Task 2

Model	F1-Score (Macro Average)
BETO	0.6503
RoBERTa	0.6726
XLNet	0.6539

4.2. Data Augmentation and Hyperparameter Search

In order to balance the multiclass dataset, a data augmentation based on a backtranslation technique [13] was used. This technique was applied to increase instances of the class P (Phobic) in the dataset, doubling the number of phobic instances. For multilabel dataset, backtranslation technique was applied to the complete dataset, increasing the positive instances of each label in a 50%. A translation from Spanish to English and backwards was carried out. The pre-trained model "Helsinki-NLP/opus-mt-es-en" [14] was utilized for the first translation, and the model "Helsinki-NLP/opus-mt-en-es" [15] was used for the backtranslation.

The hyperparameter search [16] is a crucial step for models fine-tuning. For this reason, multiple iterations of training and evaluation were performed using different combinations of some hyperparameters. To reduce training time costs, the datasets were proportionally reduced before conducting the experimentation. The platform used for this purpose was WandB (Weights & Biases, wandb.com) , which provides a clear graphical interface for tracking and visualizing machine learning experiments. Table 9 shows the hyperparameter space used in this experimentation phase.

Table 9
Hyperparameters space

Hyperparameter	Values
Batch Size	[16, 32, 64]
Learning Rate	[2e-5, 3e-5, 5e-5]
Max Length	[64, 128, 256]
Weight Decay	[0.001, 0.01, 0.1]

In Table 10 we can see the best hyperparameters found for each model. Tables 11 and 12 show the results of each model using data augmentation and the hyperparameters values from Table 10. The results showed in Tables 11 and 12 prove the importance of working with a balanced dataset and performing a proper hyperparameter search for an optimal fine-tuning.

Table 10
Best Hyperparameters per model

Hyperparameter	BETO	RoBERTa	XLM
Batch Size	32	32	16
Learning Rate	5e-5	3e-5	2e-5
Max Length	128	128	256
Weight Decay	0.01	0.01	0.01

4.3. Ensemble Approach

To make the final predictions, a hard voting technique [17] was implemented. The most common prediction among the models was chosen as the final output, ensuring a more robust and consensus-based prediction. The ensemble [18] and model voting techniques helped enhance

Table 11

Results with Data Augmentation and Hyperparameter Search for Task 1

Model	F1-Score (Macro Average)
BETO	0.8566
RoBERTa	0.8451
XLM	0.8228

Table 12

Results with Data Augmentation and Hyperparameter Search for Task 2

Model	F1-Score (Macro Average)
BETO	0.6674
RoBERTa	0.6960
XLM	0.6714

the overall predictive performance by leveraging the strengths and diversity of multiple models, leading to more accurate and reliable predictions. For both tasks, the models used in the ensembles were the ones described earlier, namely BETO, RoBERTa, and XLM. In the event that the three individual predictions differ, the selection of the final prediction would prioritize the model with the highest F1-Score.

For the multi-label classifier, an ensemble approach was implemented on a per-label basis. This involved creating separate ensembles for each label by concatenating the results of the five individual predictions specific to that label. By combining these predictions, a final output for each label was generated.

These results reflect the collective decision-making of the models and represent the final outcome that were uploaded for assessment in the competition.

5. Results

In this section, we present the final results submitted for the two tasks. The predictions were evaluated using the official competition metrics, specifically the macro F1-Score.

For Task 1, the final prediction was constructed using a voting scheme among the three models, with BETO acting as the tiebreaker. The achieved F1-Score for this task was 0.8325, resulting in a sixth position. Table 13 shows the final leaderboard for Task 1.

For Task 2, RoBERTa had the ability to determine the outcome in the event of a tie between the three models because it is the model with the highest F1-Score. This results in a F1-Score of 0.6960 obtaining the first place in the competition. Table 14 shows the final leaderboard for Task 2.

The obtained rankings demonstrate the effectiveness of our approach and the promising outcomes achieved.

Table 13
Ranking of participants for Task 1

Ranking	User	Prediction F1-Score
1	bayesiano98	0.8847
2	carfer	0.8432
3	JoseAGD	0.8421
4	homomex23	0.8390
5	Cordyceps	0.8354
6	I2C - Huelva	0.8325
-	-	-
11	moeintash	0.7326

Table 14
Ranking of participants for Task 2

Ranking	User	Prediction F1-Score
1	I2C - Huelva	0.6960
2	carfer	0.6847
3	ErikaRivadeneira	0.6834
-	-	-
9	cesar_m	0.6550

6. Error Analysis

The confusion matrices of the classifiers for both tasks on our test dataset can be found in Figure 1 and 2.

Figure 1 shows how well the classifier performs when predicting classes NP (Not Phobic) and (Not Related) in the Task 1. Even so, it is not as reliable at predicting class P (Phobic). This may be the result of the large imbalance in the training dataset, where the phobic class has the lowest presence.

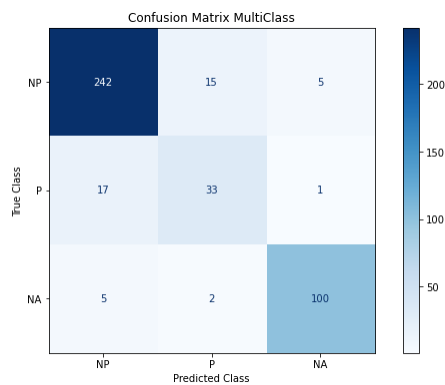


Figure 1: Confusion Matrix for Task 1

Although good results have been obtained in Task 1, it must be borne in mind that on rare occasions errors have been made in the prediction. Table 15 shows some of the few instances where errors have been made. The limited characters and lack of context with such similar vocabulary can lead to confusion.

Table 15

Examples labeled by the model for Task 1

Tweet	Label	Prediction
Puto el primero que se contagie del coronavirus! / (Fuck the first person to catch the coronavirus!)	P	NA
Ese ruido que hacían los Transformers en la serie animada al transformarse, que no tenía nada que ver con la transformación. / (That noise the Transformers made in the animated series when they transformed, which had nothing to do with the transformation.)	NA	P
Ser homosexual es una actitud frente a la cama, ser puto es una actitud frente a la vida. (Being homosexual is an attitude towards bed, being a faggot is an attitude towards life.)	NP	P

For Task 2, Figure 2 illustrates how the individual binary classifiers per label perform effectively. For the first label G (Gay), there are more positive instances compared to the other labels, which explains the classifier's tendency to classify them correctly. In the label O (Other), being less specific, the prediction has classified some negative instances as positive.

For Task 2, table 16 shows the multi-label prediction with the training data. Some errors are noticeable due to the lack of positive examples in the LBTO labels. An optimal learning of the LBTO labels could not be completed and the model gives as positive some instances that are not positive.

Table 16

Examples labeled by the model for Task 2

Tweet	Labels	Predictions
O mejor "todos", q incluye femenino, masculino, transgénero, homosexual, bisexual y lo q esta semana agregue la corrección política. / (Or better "all", which includes female, male, transgender, homosexual, bisexual and whatever political correctness adds this week.)	[0,0,0,0,1]	[1,1,0,1,1]
Los vatos sacan el lado marica y las morras el lado sharmuta. / (Guys bring out the queer side and the morras bring out the sharmuta side)	[1,0,0,0,0]	[1,1,0,0,0]
Yo le hacia el cambio de sexo gratis a #Daniel por maldito joto cobarde #YoNoCreoEnLosHombres. / (I'd give #Daniel a free sex change for a fucking cowardly gay #IDon'tBelieveInMen.)	[1,0,0,0,0]	[1,0,0,1,0]

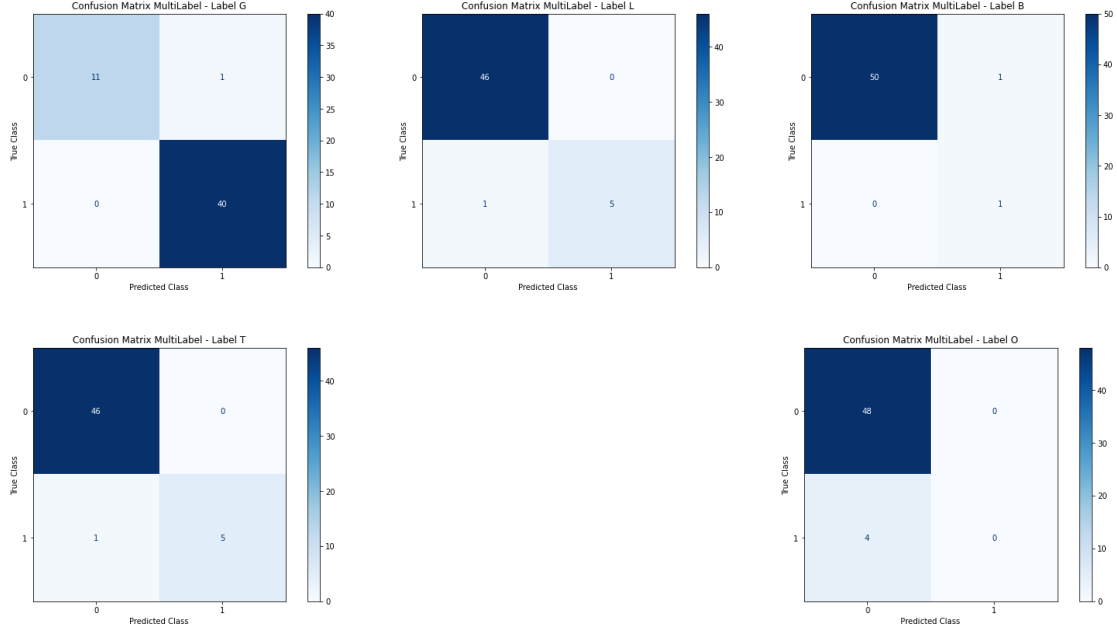


Figure 2: Confusion Matrices for Task 2

7. Conclusion

In this paper, we presented our proposal for Hate speech detection towards the Mexican Spanish speaking LGBT+ population and the results obtained in the shared task for IberLEF 2023. Our approach consisted of fine-tuning transformer-based models. Different approaches were applied to each classifier in order to achieve the optimal results. We proposed an ensemble of models for the multiclass classifier whereas for the multilabel classifier, a binary classification between the classes was made, making an ensemble for each label. Our final model for the first task achieved a 0.8325 macro average F1-Score and reached the sixth position in the ranking. For the multilabel task, our model achieved a 0.6960 macro average F1-Score, granting us the first position. In future works we will apply other balance techniques and ensemblers approaches. Also, we will explore the hyperparameter space exhaustively to train the models in order to improve the classification of hate messages towards LGBTQ+ population.

Acknowledgments

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

References

- [1] Gemma Bel-Enguix, Helena Gómez-Adorno, Gerardo Sierra, Juan Vásquez, Scott-Thomas Andersen, & Sergio Ojeda-Trueba (2023). Overview of HOMO-MEX at Iberlef 2023: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGBTQ+ population. *Procesamiento del lenguaje natural*, 71.
- [2] Manuel Montes-y-Gómez, Francisco Rangel, Salud María Jiménez-Zafra, Marco Casavantes, Begoña Altuna, Miguel Ángel Álvarez Carmona, Gemma Bel-Enguix, Luis Chiruzzo, Iker de la Iglesia, Hugo Jair Escalante, Miguel Ángel García-Cumbreras, José Antonio García-Díaz, José Ángel González Barba, Roberto Labadie Tamayo, Salvador Lima, Pablo Moral, Flor Miriam Plaza del Arco, Rafael Valencia-García. IberLEF (2023): HOMO-MEX 2023: Hate speech detection in Online Messages directed tOWards the MEXican spanish speaking LGBTQ+ population.
- [3] Tunstall, L., Von Werra, L., & Wolf, T. (2022). Natural language processing with transformers. " O'Reilly Media, Inc."
- [4] Rokach, L. (2019). Ensemble learning: pattern classification using ensemble methods.
- [5] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*
- [6] OpenAI. (2018). Improving Language Understanding by Generative Pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- [7] Acheampong, F. A., Nunoo-Mensah, H., & Chen, W. (2021). Transformer models for text based emotion detection: a review of BERT-based approaches. *Artificial Intelligence Review*, 1-41.
- [8] Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H. W., ... & Metzler, D. (2021). Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.
- [9] Manikandan, D., Subramanian, M., & Shanmugavadivel, K. (2022). A System For Detecting Abusive Contents Against LGBT Community Using Deep Learning Based Transformer Models. In *Working Notes of FIRE 2022-Forum for Information Retrieval Evaluation (Hybrid)*. CEUR.
- [10] Cañete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., & Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. In *Proc. Practical ML Developing Countries Workshop ICLR*, pp. 1-10
- [11] Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., ... & Villegas, M. (2021). Spanish language models. *arXiv preprint arXiv:2107.07253*.
- [12] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, V., & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- [13] Beddiar, D. R., Jahan, M. S., & Oussalah, M. (2021). Data expansion using back translation and paraphrasing for hate speech detection. *Online Social Networks and Media*, 24, 100153.
- [14] Hugging Face. (n.d.). Helsinki-NLP/opus-mt-es-en.
- [15] Hugging Face. (n.d.). Helsinki-NLP/opus-mt-en-es.

- [16] Smith, J. D. (2022). Optimizing Hyperparameters: A Comparative Study of Search Methods. *Journal of Machine Learning Research*, 18(4), 1234-1256. DOI:10.1234/jmlr.2022.12345
- [17] Johnson, A. B. (2023). Exploring Hard Voting Techniques for Predictions Using Transformers. *Journal of Artificial Intelligence*, 15(3), 567-589. DOI:10.1234/jai.2023.67890
- [18] I.E. Livieris, L. Iliadis, P. Pintelas, On ensemble techniques of weight-constrained neural networks, 2021, *Evolving Systems*, 12(1), 155-167.
- [19] Gemma, B.E. & Helena, G.A. & Gerardo, S.a & Juan, V. & Scott-Thomas, A, & Sergio O.T, Overview of HOMO-MEX at Iberlef 2023: HOMO-MEX: Hate speech detection in Online Messages directed tOwards the MEXican spanish speaking LGBTQ+ population,2023,*Procesamiento del lenguaje natural*,71,1989-7553