MASTER EN INGENIERIA Y CIENCIA DE DATOS

UNED 2022/2023

DETECCIÓN DE POSTURA EN CONJUNTOS DE DATOS MULTILINGÜE

Autor: Jorge Pablo Ávila Gómez

Director: Álvaro Rodrigo Yuste
Codirector: Roberto Centeno Sánchez

Índice

Resumen

La detección de postura en el aprendizaje automático es la tarea de identificar automáticamente la actitud expresada en un texto hacia una proposición específica. Esto es fundamental para el procesamiento del lenguaje natural, ya que proporciona información valiosa sobre las opiniones de las personas hacia diferentes propuestas. Puede tener aplicaciones importantes en diversos campos, desde la gestión de la reputación en línea hasta la toma de decisiones políticas. Esta tarea se lleva a cabo mediante algoritmos de minería de texto y aprendizaje automático, que analizan características clave en el texto para determinar su postura. Sin embargo, la mayoría de las investigaciones se han centrado en el inglés, lo que limita la aplicabilidad de los modelos a otros idiomas. Algunos estudios han abordado esta limitación mediante conjuntos de datos en diferentes idiomas, pero se han enfrentado a desafíos debido a desequilibrios en clases y a grandes diferencias entre los contextos de cada idioma. En esta línea, se presentó la tarea de evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*. Su objetivo es clasificar la postura de textos en varios idiomas y con respecto a múltiples objetivos. La tarea implica clasificar comentarios sobre propuestas políticas como de apoyo, oposición o neutralidad. El conjunto de datos se construyó utilizando la plataforma de la Conferencia sobre el Futuro de Europa (CoFE), que incluye propuestas políticas y comentarios en cualquiera de los 24 idiomas oficiales de la Unión Europea. Este conjunto de datos presenta una oportunidad atractiva para estudiar la detección de postura en un entorno verdaderamente multilingüe.

Este trabajo presenta los resultados obtenidos en esta tarea. El objetivo principal de este estudio fue desarrollar e investigar diversos modelos de aprendizaje automático y aprendizaje profundo para la clasificación de la postura. Se exploraron técnicas como el *transfer learning*, *data augmentation* y *label spreading*, para ampliar el conjunto de datos de entrenamiento y aprovechar modelos preentrenados. Los resultados muestran que las estrategias empleadas

en este trabajo mejoran significativamente los resultados de referencia proporcionados por la tarea. Estos resultados, después de su presentación en la tarea, se han publicado como parte de las actas de CLEF 2023 (Avila et al., 2023).

Palabras clave

Detección de postura, procesamiento del lenguaje natural, aprendizaje automático, multilingüe

1. Introducción

La tarea de detección de postura, en el contexto del aprendizaje automático, consiste en la detección automática de la actitud de un comentario, o cualquier otro tipo de texto sobre una propuesta específica. En este contexto, se busca determinar la actitud expresada en un comentario con respecto a una propuesta en particular, ya sea de apoyo, oposición o neutralidad (Mohammad et al., 2016b). Esta tarea juega un papel crucial en el procesamiento de lenguaje natural, ya que permite comprender la opinión o actitud de los individuos hacia propuestas específicas y, por lo tanto, proporciona información valiosa para la toma de decisiones informadas en diversos ámbitos, como el político, empresarial y social. El proceso de clasificación de postura se realiza a través de algoritmos basados en técnicas de minería de texto y aprendizaje automático. Estos algoritmos extraen características relevantes del texto y las utilizan para determinar la postura del mismo.

La detección de postura ha sido una herramienta relevante para las empresas y organizaciones, ya que les permite tomar decisiones informadas para mejorar sus productos o servicios. Por ejemplo, para la detección de desinformación (Hardalov et al., 2021a), u opiniones sobre productos de interés en debates en línea como *iPhone vs BlackBerry* (Somasundaran & Wiebe, 2009). Por tanto, es una herramienta útil para gestionar la reputación en línea, e identificar y comprender las opiniones y percepciones del público sobre determinados temas.

Asimismo, la clasificación de postura también es relevante en el ámbito político y en la toma de decisiones públicas. En la bibliografía podemos encontrar ejemplos de interés público como la verificación de encuestas (Joseph et al., 2021) y proyectos de consulta ciudadana a gran escala (Barriere, Balahur, et al., 2022). También encontramos otros ejemplos de análisis de debates en línea, sobre temas de justicia social como *Aborto* o *Derechos LGBTQ+* (Somasundaran & Wiebe, 2010). Al analizar grandes cantidades de datos de opiniones y comentarios, se pueden obtener información y conocimientos que ayuden a

comprender las percepciones y preferencias del público en relación con diversos temas.

Los diferentes estudios que encontramos en la bibliografía presentan diferentes formas de abordar el problema de la detección de posturas, por ejemplo, los podemos clasificar según el contexto. Entendiendo por contexto, el elemento de referencia sobre el que se está tomando una postura. Normalmente el contexto puede ser un tema, como "la inmigración", o pequeñas afirmaciones o textos. Por un lado, es muy común que todos los textos a clasificar se refieran a un mismo contesto (Mohammad et al., 2016b). También existe la posibilidad de que haya varios contextos diferentes dentro de la misma tarea, y un grupo de textos o comentarios para cada uno de los diferentes contextos[1]. Por otro lado, y mucho menos común, existen tareas que utilizan un contexto dinámico, que sería por ejemplo, cuando en un hilo de comentarios el contexto de cada comentarios son los comentarios anteriores en el hilo (Gorrell et al., 2019).

Una limitación importante en el campo de la detección de posturas es que la mayoría de los trabajos son realizadas principalmente con conjunto de datos en inglés, esto dificulta la generalización de los resultados a otros idiomas. Podemos resaltar algunos trabajos realizados en otros idiomas como árabe (Baly et al., 2018), checo (Hercig et al., 2017), francés (Evrard et al., 2020), ruso (Vychegzhanin & Kotelnikov, 2019) o italiano (Cignarella et al., 2020). Otros trabajos han intentado abordar la tarea de detección de postura desde un enfoque multilingüe. Por ejemplo, encontramos tareas con conjuntos de datos en español y catalán (Taulé et al., 2018); o francés e italiano (Lai et al., 2020). Como consecuencia de la dificultad de construir estos conjuntos de datos en diferentes idiomas, estos suelen presentar ciertas limitaciones, como clases o contextos desbalanceados entre los idiomas, lo cual dificulta la generalización de los modelos y resultados.

[1] http://www.fakenewschallenge.org/

En este contexto surge la tarea de evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*[2] que tiene como objetivo aplicar técnicas de procesamiento de lenguaje natural y aprendizaje automático para clasificar la postura de textos en diferentes idiomas y con respecto a múltiples objetivos. La tarea consiste en la clasificación de comentarios a propuesta de interés político como a favor, en contra o neutral. El conjunto de datos se ha construido usando la plataforma CoFE (*Conference on the Future of Europe*)[3] que cuenta con propuestas políticas y comentarios a estas que pueden estar escritos en cualquier de los 24 idiomas de la Unión Europea. Esto hace que podamos encontrar diferentes idiomas comentando una misma propuesta. Este corpus es, en definitiva, una opción muy atractiva para el estudio generalizado de la detección de postura en un verdadero entorno multilingüe (Barriere, Jacquet, et al., 2022).

En este trabajo, se describen los resultados obtenidos en la participación de la tarea de evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*. El objetivo principal de este trabajo ha sido el desarrollo y estudio de diferentes modelos basados en aprendizaje automático y aprendizaje profundo para la clasificación de la postura. Para ello, se han utilizado y analizado el uso de diferentes técnicas como *transfer learning*, *data augmentation* y *label spreading*, las cuales han permitido ampliar el conjunto de datos útiles para el entrenamiento y la utilización de grandes modelos preentrenados. Los resultados obtenidos muestran que las estrategias abordadas en este trabajo son capaces de mejorar significativamente los resultados propuestos como línea base por la tarea. Dichos resultados, tras ser presentados en la tarea, han sido publicados como *proceeding* CLEF 2023[4] (Avila et al., 2023).

[2] https://touche.webis.de/clef23/touche23-web/multilingual-stance-classification.html
[3] https://futureu.europa.eu/?locale=en
[4] http://www.clef-initiative.eu/

2. Objetivos

El objetivo principal de este trabajo fin de máster es el desarrollo de diferentes métodos basados en aprendizaje automático para la clasificación de la postura, y la presentación de los resultados obtenidos en la tarea de evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*. Podemos dividir el objetivo principal en diferentes subobjetivos:

- Realizar un estudio exploratorio de los diferentes conjuntos de datos y estudiar sus características más relevantes.
- Diseñar y desarrollar diferentes estrategias de modelos de clasificación de postura basado en aprendizaje automático. En este punto se van a abordar:
    - Ampliación del conjunto de entrenamiento mediante diferentes técnicas.
    - El uso de modelos preentrenados mediante *transfer learning*.
- Entrenar y evaluar el modelo de clasificación de postura, ajustando los parámetros y optimizando el rendimiento del modelo.
- Realizar un análisis detallado de los resultados obtenidos y comparación entre los diferentes enfoques.
- Presentar los resultados obtenidos en la tarea de evaluación y analizar los resultados obtenidos en la competición.

3. Estado del Arte

En este capítulo se van a abordar las definiciones y conceptos teóricos claves para entender el trabajo desarrollado en este proyecto. Primero se van a abordar las definiciones de aprendizaje automático y aprendizaje profundo, y su relación con tareas de procesamiento del lenguaje natural. Prestando especial atención en la detección de postura en textos. Luego se van a abordar las estrategias que existen para convertir textos en datos que puedan manejar los modelos de aprendizaje. Se continuará detallando las técnicas más utilizadas para aumentar la cantidad de datos disponibles para los entrenamientos en tareas de detección de posturas las técnicas utilizadas que se han utilizado en este trabajo que son *data augmentation* y *label spreading*. También se explicará en que consiste la técnica del *transfer learning* y los modelos pre-entrenados. El uso de este tipo de modelos es una de las estrategias más utilizadas en el ámbito del procesamiento del lenguaje natural y la detección de postura como detallaremos más adelante. trabajo.

3.1. Aprendizaje automático y aprendizaje profundo

El aprendizaje automático y el aprendizaje profundo son dos disciplinas fundamentales en el campo de la inteligencia artificial, que han revolucionado la forma en que las máquinas pueden aprender y realizar tareas complejas. Ambos enfoques se basan en la idea de entrenar modelos computacionales para que adquieran conocimientos y realicen predicciones o clasificaciones precisas.

El **aprendizaje automático**, también conocido como *machine learning*, se refiere a la capacidad de las máquinas para aprender de manera automática a partir de los datos sin ser explícitamente programadas. En lugar de seguir instrucciones específicas, los algoritmos de aprendizaje automático se diseñan para detectar patrones y tomar decisiones basadas en la información disponible.

El **aprendizaje profundo**, también conocido como *deep learning*, es una rama del aprendizaje automático que se inspira en la estructura y funcionamiento del cerebro humano. Utiliza redes neuronales artificiales para simular el procesamiento de información en capas sucesivas de nodos interconectados. Estas redes neuronales profundas son capaces de aprender representaciones jerárquicas de los datos, lo que les permite capturar características complejas y sutiles de los patrones presentes en los conjuntos de datos.

Ambos enfoques, el aprendizaje automático y el aprendizaje profundo, son ampliamente utilizados en tareas de clasificación. La clasificación implica asignar una etiqueta o una categoría a un determinado conjunto de datos, y puede ser útil en una variedad de campos, como la medicina, la publicidad en línea, el análisis de sentimientos y la detección de fraudes, entre otros. Tanto el aprendizaje automático como el aprendizaje profundo ofrecen métodos poderosos para abordar estas tareas, permitiendo a las máquinas realizar clasificaciones precisas y automatizadas en grandes volúmenes de datos.

El **aprendizaje automático** ha sido ampliamente utilizado en la clasificación de textos. Los enfoques tradicionales, como los algoritmos de aprendizaje supervisado, utilizan técnicas estadísticas para extraer características relevantes de los textos, como la frecuencia de las palabras o la presencia de ciertos patrones. Estas características se utilizan para entrenar modelos de aprendizaje automático, como los clasificadores bayesianos ingenuos o las máquinas de vectores de soporte, que pueden asignar categorías a nuevos textos en función de las características aprendidas.

Sin embargo, el **aprendizaje profundo** ha revolucionado la clasificación de textos al introducir redes neuronales artificiales. Estas redes neuronales profundas son capaces de aprender representaciones de texto de manera jerárquica, capturando características complejas y abstractas. Por ejemplo, se pueden utilizar redes neuronales convolucionales para extraer características locales de los textos, como las combinaciones de palabras, mientras que las

redes neuronales recurrentes pueden capturar dependencias a largo plazo en las secuencias de palabras.

Al entrenar estas redes neuronales con grandes cantidades de datos etiquetados, los modelos de aprendizaje profundo pueden aprender automáticamente a reconocer patrones sutiles en los textos y realizar clasificaciones precisas. Estos modelos han demostrado un rendimiento sobresaliente en diversas tareas de clasificación de textos, superando en muchos casos los enfoques tradicionales de aprendizaje automático. Además, el aprendizaje profundo ha permitido el desarrollo de modelos de clasificación de textos que pueden aprender características y representaciones directamente de los datos sin necesidad de una ingeniería de características manual. Esto significa que los modelos de aprendizaje profundo pueden adaptarse de manera más flexible a diferentes tipos de textos y capturar la complejidad inherente a los lenguajes naturales.

3.2. Procesamiento del lenguaje natural

En el campo del aprendizaje automático y la inteligencia artificial, uno de los desafíos más importantes es la capacidad de convertir los textos en datos que puedan ser procesados y utilizados por los modelos de aprendizaje. Los textos son una forma de información no estructurada, lo que significa que no se presentan en un formato fácilmente interpretable para las máquinas. Sin embargo, existen diversas estrategias y técnicas que permiten transformar los textos en datos estructurados, facilitando su análisis y utilización en los modelos de aprendizaje.

Una de las estrategias más comunes para convertir textos en datos es el proceso de **tokenización**. La tokenización implica dividir el texto en unidades más pequeñas llamadas tokens, que pueden ser palabras individuales o incluso caracteres. Al dividir el texto en tokens, se crea una representación estructurada que puede ser procesada por los modelos de aprendizaje. Además, la

tokenización también puede incluir la eliminación de signos de puntuación, caracteres especiales o palabras vacías para limpiar el texto y reducir el ruido en los datos.

Otra estrategia importante es la representación vectorial de los textos. Esto implica asignar un vector numérico a cada token o palabra en el texto. Hay diferentes enfoques para lograr esto, como el modelo de bolsa de palabras (*bag-of-words*) o el modelo de *embeddings*. En el modelo de bolsa de palabras, cada token se representa mediante un vector de características que indica su presencia o frecuencia en el texto. Por otro lado, los *embeddings* son representaciones vectoriales más densas y contextualizadas que capturan el significado semántico de las palabras en función de su contexto.

3.3. Detección de postura

La Detección de Postura es una tarea importante dentro del Procesamiento del Lenguaje Natural que tiene como objetivo identificar la postura o actitud del autor de un texto hacia un contexto específico. El contexto puede ser una persona, una organización, una política, etc. La postura del autor se puede categorizar como favorable, contraria o neutral hacia el objetivo (Küçük & Can, 2021b).

Los primeros trabajos sobre detección de postura han utilizado diferentes algoritmos basados en aprendizaje automático en una variedad de géneros de textos, incluyendo debates políticos, foros de debate en línea, ensayos de estudiantes o *tweets* (Anand et al., 2011; Faulkner, 2014; Hasan & Ng, 2013; Rajadesingan & Liu, 2014; Thomas et al., 2006). En estos trabajos se han empleado tanto enfoques tradicionales de aprendizaje automático como máquinas de vectores de soporte (SVM) (Küçük & Can, 2018) y regresión logística (Cignarella et al., 2020), como enfoques de aprendizaje profundo como redes neuronales recurrentes (Dey et al., 2018; Sun et al., 2019) o redes neuronales convolucionales (Vijayaraghavan et al., 2016; Wei et al., 2016; Zhou et al., 2017) para la tarea de detección de postura. Siendo cada vez más común

la aplicación de los algoritmos de aprendizaje profundo en los estudios más recientes (Küçük & Can, 2021a).

Existen importantes esfuerzos en el campo por crear conjuntos de datos anotados para la detección de postura (Küçük & Can, 2019; Lai et al., 2020). Con respecto a los idiomas de los conjuntos de datos anotados podemos encontrar el inglés (Mohammad et al., 2016a; Sobhani et al., 2017), catalán (Taulé et al., 2017), chino (Xu et al., 2016), checo (Hercig et al., 2017), italiano (Lai et al., 2018), español (Taulé et al., 2017) o turco (Küçük & Can, 2019). Además de estos conjuntos de datos específicos para cada idioma, recientemente se están compilando conjuntos de datos anotados multilingües (Lai et al., 2020; Vamvas & Sennrich, 2020; Zotova et al., 2021). En el contexto de este trabajo, la tarea de evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*, propone un nuevo conjunto de datos multilingüe parcialmente anotado (Barriere, Jacquet, et al., 2022). A pesar del considerable progreso, la escasez de datos de entrenamiento etiquetados es uno de los principales desafíos, debido al complejo y tedioso trabajo de anotación. Este desafío lo encontramos particularmente en la tarea *Intra-Multilingual Multi-Target Stance Classification 2023*, donde en el conjunto de datos para la tarea solo hay anotados con las tres clases 1400 comentarios de los más de 21000 comentarios proporcionados.

Para superar este problema, en la bibliografía se han propuesto diferentes técnicas, entre las más utilizadas se encuentran el *data augmentation*, el *label spreading* y el *transfer learning*.

El *data augmentation* es una técnica para aumentar el tamaño y la diversidad del conjunto de entrenamiento mediante la creación de nuevos ejemplos de datos sintéticos. En el contexto de la detección de postura, el *data augmentation* se puede lograr generando muestras sintéticas a partir de los datos originales, por ejemplo, mediante traducción, que contengan objetivos y posturas similares a los datos de partida.

El *label spreading* es una técnica de aprendizaje semisupervisado donde la idea es propagar las etiquetas desde los datos etiquetados hacia los datos no

etiquetados. Esta técnica puede ser útil en el contexto de la detección de postura cuando tenemos una pequeña cantidad de datos etiquetados y una gran cantidad de datos no etiquetados.

El *transfer learning* es una técnica de aprendizaje automático donde se utiliza un modelo pre-entrenado en un problema nuevo y similar. En el contexto de la detección de postura, el *transfer learning* se puede utilizar para transferir el conocimiento de modelos entrenados en otras tareas de procesamiento del lenguaje natural a el problema de la detección de postura.

3.4. *Data Augmentation*

El ***data augmentation*** es una técnica utilizada para aumentar la cantidad y la diversidad de los datos de entrenamiento. Consiste en aplicar transformaciones o modificaciones a los datos existentes con el fin de generar nuevas instancias que sean diferentes pero que mantengan la misma información o etiquetas. El objetivo principal del ***data augmentation*** es mejorar el rendimiento y la generalización de los modelos de aprendizaje automático al proporcionarles más ejemplos variados y realistas para aprender.

En el contexto del procesamiento del lenguaje natural estas serían algunas de las técnicas más comunes de ***data augmentation***:

1. Sinónimos: Reemplazar algunas palabras por sus sinónimos puede ayudar a generar variantes del texto original.
2. Inserción y/o eliminación de palabras: Agregar o eliminar palabras en puntos aleatorios del texto original puede aumentar la variabilidad y obligar al modelo a confiar en otras señales contextuales para comprender el significado del texto.
3. **Traducción**: Utilizar traducción automática para generar variantes del texto original en diferentes idiomas.

La técnica de traducción descrita en el punto 3 fue utilizada reciente por Barriere et al. (Barriere & Balahur, 2020) para expandir un conjunto de datos en inglés (tweets en inglés), mediante traducción a otros idiomas. El entrenamiento con ellos les permitió expandir y mejorar sus resultados en tweets escritos en idiomas diferentes al inglés.

En el contexto de este trabajo, se ha utilizado de manera similar la traducción para expandir el conjunto de entrenamiento. Sin embargo, en este caso después de la traducción a otro idioma, se ha vuelto a traducir al idioma original. Este proceso introduce pequeñas variaciones a los textos, que no afectan al significado y a la vez mantiene el idioma original. Por tanto, conseguimos expandir el conjunto de entrenamiento, manteniendo la proporción original de los diferentes idiomas.

3.5. *Label Spreading*

El ***label spreading*** es una técnica de aprendizaje automático semi-supervisado que se puede utilizar en tareas del procesamiento del lenguaje natural para propagar etiquetas conocidas en datos no etiquetados. El objetivo principal del *label spreading* es utilizar la información disponible en los datos etiquetados para asignar etiquetas a datos no etiquetados.

El *label spreading* es particularmente útil en situaciones donde hay un conjunto limitado de datos etiquetados pero una gran cantidad de datos no etiquetados disponibles. Esta técnica se ha aplicado en trabajos anteriores relacionados con la detección de postura, por ejemplo, Hardolav et al, utilizan una combinación de *label spreading* y adaptación de dominios para ajustar de forma no supervisada las etiquetas de 16 conjuntos de datos diferentes (Hardalov et al., 2021b). Otro ejemplo de uso relacionado es la propagación de etiquetas en tareas de detección de comunidades donde los individuos se agrupan en nodos y la propagación se realiza considerando alguna medida de similitud entre los nodos (X. Chen & Zhao, 2022; Patel & Verma, 2023). Aunque no aborda directamente

la detección de postura, las técnicas utilizadas en estos trabajos se pueden adaptar para tareas de detección de postura, donde nodos similares es posible que mantenga una misma postura.

3.6. *Transfer Learning*

El ***transfer learning*** es una técnica que consiste en utilizar el conocimiento adquirido en una tarea general o fuente para mejorar el rendimiento en otra tarea objetivo relacionada. En lugar de entrenar un modelo desde cero para cada tarea específica, se aprovecha el conocimiento previo de un modelo pre-entrenado en una tarea relacionada para iniciar el aprendizaje en la tarea objetivo.

En el procesamiento del lenguaje natural, el *transfer learning* se ha vuelto ampliamente utilizado debido a la existencia de modelos de lenguaje pre-entrenados de gran escala, como BERT o GPT. Estos modelos se entrenan con grandes cantidades de datos para aprender representaciones de lenguaje generalizadas que capturan conocimiento lingüístico y contextual.

Una vez que el modelo de lenguaje pre-entrenado ha capturado el conocimiento general del lenguaje, se realiza un “ajuste fino” en la tarea objetivo utilizando un conjunto de datos más pequeño y etiquetado. El modelo pre-entrenado se "descongela" y se permite que los pesos se ajusten durante el entrenamiento en la tarea objetivo. En esta etapa, los pesos se adaptan a los datos específicos de la tarea objetivo, lo que permite al modelo capturar características y patrones relevantes para esa tarea en particular. *Transfer learning* fue usado para obtener la puntuación más alta en la tarea *SemEval-2016 Task 6* sobre la detección de postura en Tweets (Zarrella & Marsh, 2016). En este otro trabajo, se realizó un ajuste fino de los modelos **BERT** (Devlin et al., 2018) y **RoBERTa** (Liu et al., 2019) para una tarea de detección de postura en un conjunto de datos de *fake news* (Slovikovskaya, 2019).

En el presente trabajo se han usado modelos pre-entrenados de tipo **BERT** (Devlin et al., 2018), **RoBERTa** (Liu et al., 2019) y **XLM-RoBERTa** (Conneau

et al., 2019) de la librería para Python *Huggingface*[5] (Wolf et al., 2019). Estos modelos han sido entrenados mediante ajuste fino con los conjuntos de datos proporcionados para la tarea, y los generados por *data augmentation* y *label spreading*. El uso de modelos pre-entrenados para la tarea ha sido muy importante ya que, cuando los datos etiquetados son escasos como en nuestro caso, aprovechar el conocimiento de los modelos pre-entrenados en tareas lingüísticas generales, permite obtener buenos resultados evitando costosos y largos tiempos de entrenamiento.

3.6.1. Características de los modelos pre-entrenados BERT, RoBERTa y XLM-RoBERTa

BERT(Devlin et al., 2018), RoBERTa(Liu et al., 2019) y XLM-RoBERTa(Conneau et al., 2019) son modelos de lenguaje pre-entrenados que se basan en la arquitectura de *Transformer* (Vaswani et al., 2017). Esta es una descripción de las principales diferencias entre ellos:

1. Arquitectura:
    - **BERT** (*Bidirectional Encoder Representations from Transformers*) (Devlin et al., 2018) es uno de los modelos de lenguaje pre-entrenados más conocidos. Se basa en la arquitectura del *Transformer* y utiliza una estructura codificador-decodificador. El modelo BERT original se entrena en dos tareas: predicción de máscara de palabras (*Masked Language Modeling*, MLM) y clasificación de secuencia siguiente (*Next Sentence Prediction*, NSP).
    - **RoBERTa** (*Robustly Optimized BERT Approach*) (Liu et al., 2019): RoBERTa es una mejora de BERT que se enfoca en optimizar los hiperparámetros y el proceso de entrenamiento. RoBERTa utiliza una arquitectura similar a BERT, pero se entrena en una configuración más grande y se ajustan ciertos aspectos del preprocesamiento de los datos de entrenamiento. Esto lleva a mejoras significativas en el rendimiento en comparación con BERT.

[5] https://huggingface.co/models

- **XLM-RoBERTa** (*Cross-lingual Masked Language Model RoBERTa*) (Conneau et al., 2019): XLM-RoBERTa es una variante multilingüe de RoBERTa. Se entrena en varios idiomas y está diseñado para comprender y generar texto en múltiples idiomas. XLM-RoBERTa es útil para tareas de procesamiento del lenguaje natural que involucran datos multilingües.

2. Número de parámetros:
   - El modelo **BERT** base tiene alrededor de 110 millones de parámetros.
   - **RoBERTa** tiene múltiples variantes, pero la versión "base" tiene aproximadamente 125 millones de parámetros.
   - Al igual que RoBERTa, **XLM-RoBERTa** también tiene múltiples variantes, pero la versión "base" tiene aproximadamente 125 millones de parámetros.
3. Conjunto de datos y entrenamiento:
   - **BERT**: Se entrenó principalmente en textos en inglés utilizando libros y artículos de Wikipedia.
   - **RoBERTa** se entrenó en un conjunto de datos mucho más grande y diverso que BERT, incluyendo textos de Wikipedia, Common Crawl y libros.
   - **XLM-RoBERTa** se entrena en varios idiomas utilizando un conjunto de datos multilingüe.

4. Descripción de la tarea y marco de la competición

4.1. La iniciativa *Touché*

La tarea se enmarca dentro de *Touché*[6], como una iniciativa de *CLEF* (*Conference and Labs of the Evaluation Forum*)[7]. *CLEF* se define en dos partes principales: laboratorios de evaluación de sistemas de acceso a la información y talleres para discutir y probar actividades de evaluación innovadoras, y una conferencia revisada por pares que abarca una amplia gama de temas relacionados con la investigación, experimentos y metodologías de evaluación en el campo del acceso a la información multilingüe y multimodal. *CLEF* proporciona una infraestructura para la prueba, ajuste y evaluación de sistemas multilingües y multimodales, así como para la investigación sobre el uso de datos no estructurados, semi-estructurados, altamente estructurados y enriquecidos semánticamente en el acceso a la información. Además, *CLEF* fomenta la creación de colecciones de pruebas reutilizables, la exploración de nuevas metodologías de evaluación y formas innovadoras de utilizar datos experimentales, y brinda un espacio para la discusión de resultados, la comparación de enfoques, el intercambio de ideas y la transferencia de conocimientos.

La organización de este tipo de tareas se lleva haciendo desde el año 2020. Otras tareas que se han desarrollado dentro de *CLEF* han sido *Argument Retrieval for Controversial Question*[8], que consistió en dado un tema controvertido, recuperar y clasificar documentos según su relevancia y calidad argumentativa, y detectar la postura del documento; o *Human Value Detection*[9], donde, dado un argumento textual y una categoría de valor humano, había que clasificar si el argumento se basa o no en esa categoría.

[6] https://touche.webis.de/
[7] http://www.clef-initiative.eu/
[8] https://touche.webis.de/clef23/touche23-web/argument-retrieval-for-controversial-questions.html
[9] https://touche.webis.de/semeval23/touche23-web/index.html

4.2. Intra-Multilingual Multi-Target Stance Classification

En la edición de CLEF 2023, *Touché* ha organizado una nueva tarea en el ámbito de la detección de postura, la tarea *Intra-Multilingual Multi-Target Stance Classification*[10]. El objetivo de esta tarea es la clasificación de comentarios a propuestas sobre tema socialmente relevantes, que se han escrito en la plataforma *Conference on the Future of Europe* (CoFE)[11]. CoFE es una plataforma en línea en la cual cualquier usuario puede escribir una propuesta en cualquiera de los 24 idiomas de la UE. Otros usuarios pueden comentar y/o respaldar una propuesta u otro comentario. Todos los textos se traducen automáticamente a cualquiera de los 24 idiomas de la UE. La plataforma cuenta con más de 20.000 comentarios sobre 4200 propuestas en 26 idiomas. El inglés, alemán y francés son los idiomas principales de la plataforma. La tarea consiste en clasificar si estos comentarios están **a favor**, **en contra** o es **neutral** hacia la propuesta. Las propuestas, los títulos y comentarios pueden estar escritos en cualquiera de los 24 idiomas oficiales de la Unión Europea (además de catalán y esperanto). La tarea proporciona una parte de los comentarios pertenecientes a esta plataforma para poder entrenar y desarrollar diferentes estrategias de aprendizaje y predicción. Una parte pequeña de los datos es guardada por la plataforma como conjunto de prueba para evaluar las predicciones que se presenten.

Los comentarios se dividen en tres conjuntos principales: **CF_S**, compuesto por 7.000 comentarios anotados por los escritores con una postura a favor o en contra; **CF_U**, que consta de 12.000 comentarios sin etiquetar; y **CF_E-Dev**, que contiene 1400 comentarios multilingües anotados por evaluadores externos en las tres categorías, a favor, en contra y neutral. Finalmente, el conjunto de datos de prueba es llamado **CF_E-Test**, que incluye 1.200 comentarios anotados por evaluadores externos, en las tres categorías. Este conjunto de test es desconocido para los participantes. Solo se puede acceder a él una vez que se va a realizar un envío de resultados, y solamente se da acceso al conjunto de

[10] https://touche.webis.de/clef23/touche23-web/multilingual-stance-classification.html
[11] https://futureu.europa.eu/?locale=en

test sin etiquetar. Esto hace que no se puedan hacer experimentos o corroboraciones utilizando este conjunto de datos, ya que los resultados son calculados externamente por la organización.

Además de los comentarios, en todos los conjuntos de datos se incluyen metadatos adicionales, como el título, el tema, el ID del escritor, el idioma y otros detalles relevantes. Junto a los conjuntos de datos con los comentarios, se incluye un conjunto de datos extra, **prop_CF**, que contiene el texto de las 4.200 propuestas a las que hacen referencia los comentarios.

Unas de las principales dificultades de esta tarea es el pequeño tamaño del conjunto de datos con las 3 etiquetas, **CF_E-Dev**. Por lo tanto, será necesario explorar diferentes alternativas para poder utilizar el resto de los comentarios pertenecientes a los otros dos conjuntos de datos.

5. Análisis exploratorio de los datos

En esta sección vamos a realizar un análisis exploratorio de los datos disponibles para la tarea. El objetivo principal de este capítulo, es describir las características principales de los diferentes conjuntos de datos, entender las similitudes y diferencias entre ellos, y explorar las posibles correlaciones que puedan aparecer entre las diferentes variables.

La Tabla 1 resume las principales características de los conjuntos de datos proporcionados. Podemos observar que solo el conjunto CF_E-D cuenta con las 3 etiquetas. Sin embargo, este es el conjunto con un menor número de elementos, y además, cuenta con solo 4 idiomas deferentes de los 24-25 en el resto de conjuntos de datos. Estas características supondrán ciertas dificultades a la hora de realizar el entrenamiento de los modelos.

En esta sección vamos a explorar los diferentes conjuntos de datos de entrenamiento y analizar las principales conclusiones que se han obtenido de ellos.

Tabla 1. Descripción de los diferentes conjuntos de datos proporcionados.

	Número de elementos	Número de etiquetas	Número de idiomas	Número de temas
Proposiciones	4247	-	24	10
CF_S	7002	2	25	10
CF_U	13213	0	25	10
CF_E-D	1414	3	4	10

5.1. Proposiciones

Cada comentario hace referencia a una proposición. Las proposiciones y los comentarios se relacionan por el *id* de la proposición. Es decir, para cada comentario vamos a tener la información del *id* de la proposición a la que pertenece. Con respecto a la información que conocemos de cada proposición, tenemos, el *título de la propuesta*, el *texto de la propuesta*, el *idioma* en el que

está escrita la propuesta, el *título traducido al inglés*, *la propuesta traducida al inglés*, el *tema de la propuesta*, y el *número de endormsements* o apoyos que ha recibido.

Tabla 2. Ejemplo de dos proposiciones con los diferentes campos que se suministran.

id	title	proposal	proposal_en	title_en	Topic	lan	endorsements
238	Renforcer Frontex et contrôler l'immigration	Renforcer Frontex et...	Strengthening Frontex...	Strengthening Frontex and controlling immigration	Migration	fr	39
224	Abschiebung unberechtigter Asylbewerber/Migranten	Alle unberechtigten Bewerber...	All unauthorised applicants...	Removal of unauthorised asylum seekers/migrants	Migration	de	45

Las propuestas están clasificadas en los siguientes temas: *'Migration', 'GreenDeal', 'Health', 'Economy', 'EUInTheWorld', 'ValuesRights', 'Digital', 'Democracy', 'Education' y 'OtherIdeas'*. Como vemos son temas de actualidad, en general importantes para las democracias europeas. Y estás propuestas están escritas en 24 lenguajes diferentes. En la Imagen 1 se muestra la distribución del número de proposiciones por tema. Vemos que hay cierto desbalanceo entre los temas, siendo *GreenDeal* y *Democracy* los que presentan una mayor prevalencia, y *Health* y *Digital*, de los que menos propuestas se hacen.

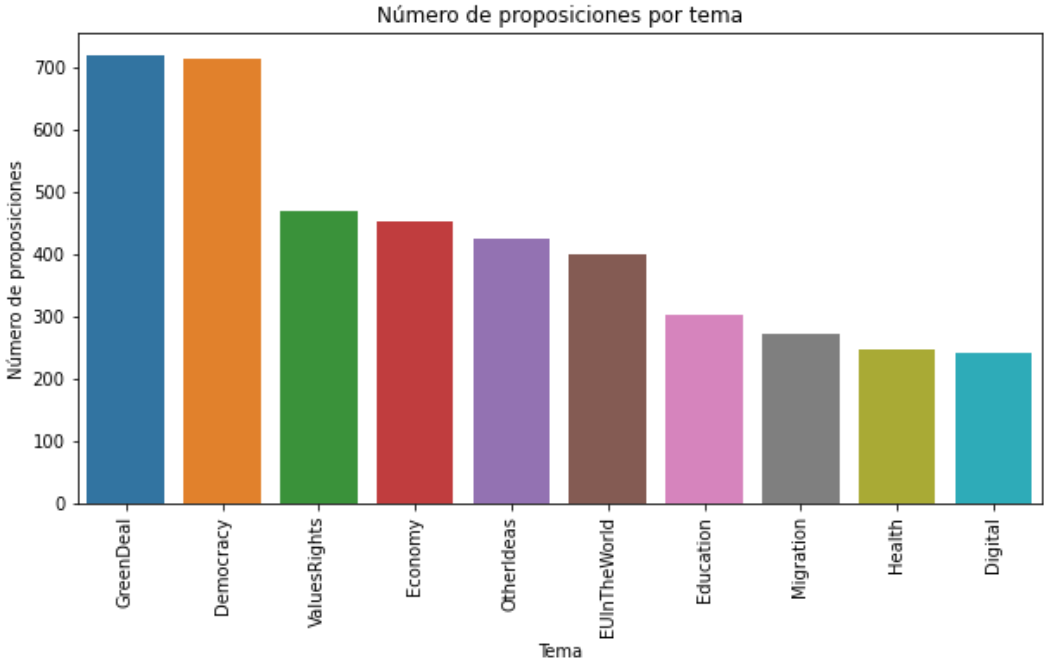

Imagen 1. Número de proposiciones por tema. *GreeDeal* y *Democracy* son los temas más populares con más de 700 proposiciones cada uno.

En la Imagen 2 podemos ver el número de proposiciones por idioma, y a pesar de haber 24 idiomas diferentes, la mayoría están escritos en inglés. Hay una presencia importante de otros idiomas como el alemán, francés, italiano y español, pero para el resto de los idiomas, el número de proposiciones es bajo.

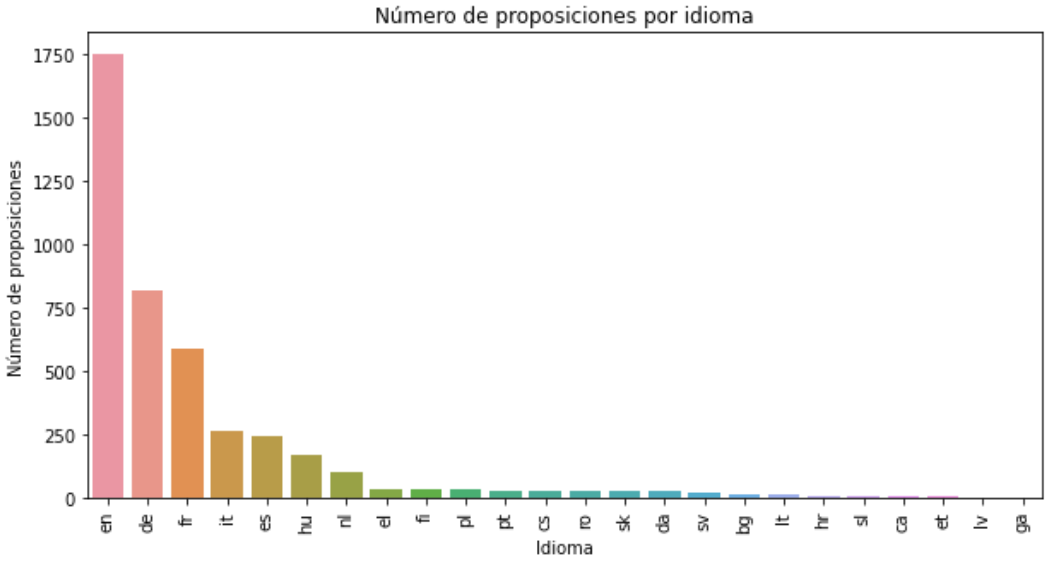

Imagen 2. Número de proposiciones por idioma. La distribución está muy desbalanceada, siendo el inglés el idioma más utilizado.

5.2. Comentarios

En la siguiente tabla (Tabla 3) vemos unos ejemplos de la estructura general de los comentarios:

Tabla 3. Ejemplos de comentarios con todos sus campos. El texto del comentario (campo *comment*) está recortado para que cupiese en la tabla.

id	id_prop	alignment	comment	depth	thread_id	last_comment_in_thread	Up vote	down vote	Topic	lan	time	label
comment_872	238	None	Frontex muss vor allem Hilfe und Unterstützung...	0.0	NaN	False	3	5	Migration	de	2021-04-21T14:55:38+02:00	Other
comment_132661	238	None	Ich würde darum bitten, die Grundrechte im Aug...	0.0	NaN	False	0	1	Migration	de	2021-11-02T23:36:33+01:00	Other

Vemos que los comentarios están identificados con un id y que mediante la columna *id_prop* los podemos relacionar con la proposición a la que pertenecen. La etiqueta del comentario la podemos encontrar en la columna *alignment* o *label*, dependiendo del conjunto de datos. Si tenemos la etiqueta en una de las columnas, en la otra aparecerá *None*. Y en el caso del conjunto **CF_U** sin etiquetar, aparece *None* en los dos campos. Además del texto del comentario encontramos otra información como el lenguaje en el que está escrito (*lan*), la hora a la que se escribió (*time*), si ha recibido votos positivos (*upvote*) o negativos (*downvote*), por último, información del hilo del comentario, su id (*thread_id*), profundidad (*depth*) y si es el último comentario en el hilo (*last_comment_in_thread*).

Con respecto a los temas, encontramos los mismos 10 temas en los 3 conjuntos de datos. Con una distribución similar en los 3 casos Imagen 3.

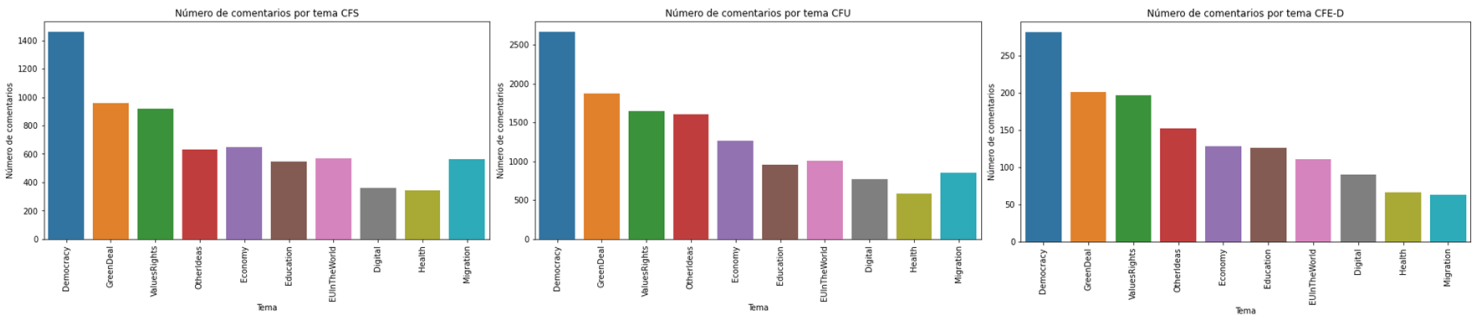

Imagen 3. Distribución del número de comentarios por tema para los conjuntos de datos, de izquierda a derecha, CFS, CFU y CFE-D.

Con respecto a los idiomas (Tabla 1), para los conjuntos **CF_S** y **CF_U**, tenemos 25 idiomas, que son los mismo que en las proposiciones, más el esperanto. Por otro lado, tenemos en el conjunto **CF_E-D** solo 4 idiomas, inglés, francés, alemán y español. Ya que este es el único conjunto de datos que tiene las 3 etiquetas, va a ser una desventaja que no tenga variedad de idiomas y, además, sea el menos numeroso.

Con respecto a la información relacionada con los hilos de los comentarios, solo es correcta en el conjunto **CF_U**, en los otros conjuntos, la información de la profundidad es siempre 0, y la información del hilo solo hace referencia a si mismo o es desconocida. El único parámetro que parece fiable es la variable booleana que indica si el comentario es el último del hilo. Ya que esta información solo se encuentra completa en el conjunto sin etiquetar **CF_U** no va a ser muy útil para la tarea y será principalmente ignorada.

En la Imagen 4 podemos ver la distribución de las clases objetivos en los dos conjuntos de datos etiquetados. Vemos que las clases están claramente desbalanceadas, sobre todo para la clase *Against,* en los dos conjuntos, y sobre todo en el conjunto **CFE-D**. Este desequilibrio de clase habrá que tenerlo en cuenta en el reparto de los datos entre los conjuntos de entrenamiento y test, para asegurarnos de que se mantengan las mismas proporciones. Además, para evitar que los modelos tengan preferencia por predecir la clase mayoritaria se

deberán usar métricas de evaluación adecuadas para el entrenamiento con clases desbalanceadas, como la puntuación macro-F1.

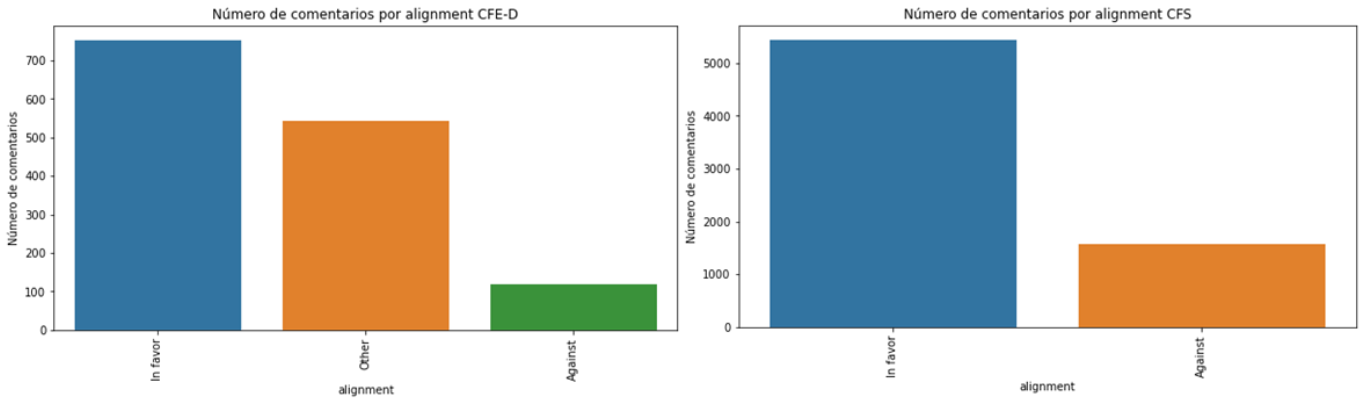

Imagen 4. Distribución del número de comentarios con respecto a la postura. Izquierda para el conjunto CFE-D. Derecha para el conjunto CFS.

5.3. Correlación entre las variables

Se ha estudiado las posibles correlaciones entre todas las variables, tanto las categóricas como las numéricas, presentes en los conjuntos de datos. Este estudio será útil para entender si las variables tabulares pueden aportar información útil para la clasificación de los comentarios. Para el estudio de las correlaciones, principalmente se han realizado representaciones gráficas de agrupaciones de las diferentes variables frente a la postura de los comentarios. A continuación, describiremos las tendencias encontradas más relevantes.

Por un lado, se ha estudiado las tendencias entre los temas y la posición de los comentarios. En este caso, como podemos observar en la Imagen 5, la distribución es relativamente estable entre los diferentes temas. Aunque resaltan los casos de la Migración que tiene una proporción alta de comentarios en contra, y la educación, con una proporción mayor de comentarios a favor. Por tanto, simplemente conociendo el tema de la propuesta nos puede ayudar a saber la tendencia general de la postura de los comentarios.

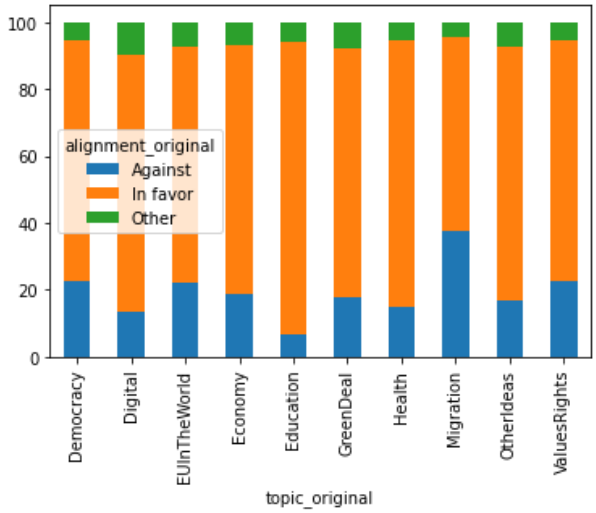

Imagen 5. Distribución en porcentaje de la postura frente a los diferentes temas.

Con respecto a los diferentes idiomas y la postura de los comentarios, no hay tendencias importantes (Imagen 6). De todos modos, podemos resaltar una ligera tendencia de los comentarios escritos en húngaro a ser en contra. Pero en general no se encuentran correlaciones importantes entre los idiomas y la postura. Además, para algunos idiomas solo hay comentarios clasificados como A favor. Esto implicaría que no es viables utilizar un modelo diferente para cada idioma, ya que no habría ejemplos para las diferentes clases e idiomas.

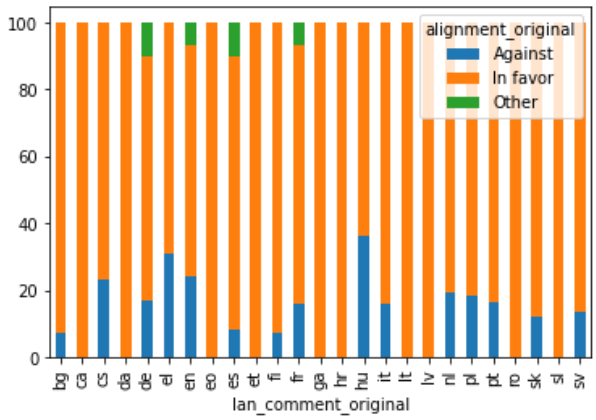

Imagen 6. Distribución en porcentaje de la postura de cada comentario frente al lenguaje en el que está escrito el comentario.

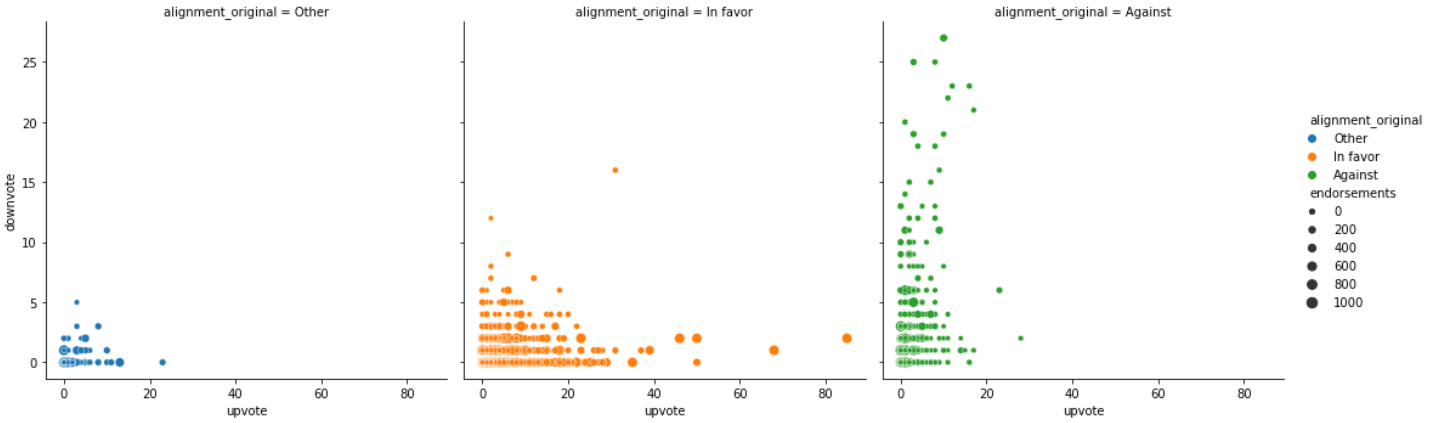

Imagen 7. Representación del número de votos negativos (*downvote*) frente al número de votos positivos (*upvote*) para cada comentario. El tamaño del punto informa acerca del *endorsement* de la propuesta a la que pertenece el comentario, un *endorsement* mayor se corresponde con un círculo más grande. Las gráficas están separadas según la postura de los comentarios, de izquierda a derecha: Neutra, A favor y En Contra.

Por último, en la Imagen 7, podemos observar las relaciones que hay entre los votos positivos y negativos que tiene cada comentario y su postura. Además, se ha añadido la información del *endorsement* en el tamaño de cada punto. Aquí, es interesante ver que se pueden distinguir ciertas características entre las diferentes posturas. Los comentarios con una postura neutra tienden a tener un número bajo de votos tanto positivos como negativos. Los comentarios con una postura a favor tienen cierta tendencia a tener un número mayor de votos positivos, sobre todo en propuesta con un alto *endorsement*. Por otro lado, para los comentarios con una postura negativa el número de votos negativos tiende a ser alto. Este análisis demuestra que la información contenida en estas variables puede ser valiosa para que los modelos pueden clasificar la postura de los comentarios.

6. Metodología

En este capítulo se van a describir las metodologías utilizadas en el trabajo. Por un lado, se van a comentar los diferentes procesamientos realizados a los conjuntos de datos para que estén en el formato adecuado para que puedan ser consumidos por los modelos de aprendizaje automático. Por otro lado, se van a explicar como se han aplicado los métodos de aumentación de los conjuntos de datos (*data augmentation* por traducción y *label spreading*). Este aumento de los datos de entrenamiento permitirá a los modelos aprender de un conjunto mayor, aumentando potencialmente su rendimiento. También se van a describir los modelos utilizados, el procedimiento de evaluación de estos y las métricas utilizadas. Por último, se comentan los modelos *baseline* propuestos para utilizarlos como referencia a la hora de comparar el rendimiento de los modelos.

6.1. Procesamiento de los comentarios

Además del texto del comentario, como hemos visto en el capítulo anterior, conocemos otras características relacionadas con el comentario y la proposición a la que pertenece. En este apartado vamos a describir las funciones que se han utilizado para añadir esta información complementaria de manera que un modelo de lenguaje natural pueda asimilarla.

La información textual del comentario se ha dividido en dos cadenas de texto. Por un lado, tenemos una cadena de texto con el contenido del comentario tal y como aparece en los datos de entrenamiento.

> *Ejemplo de Comentario: "Corruption is a root cause for many issues. And it only gets worse if the EU gives any sort of respect or legitimacy to kleptocrats. If we're going to put pressure on countries for not doing enough dirty work by stopping migrants, let us at least do it right by choosing somewhat respectable*

leaders to talk with. Not to mention being blackmailed with migrants is downright embarrassing."

Por otro lado, se ha construido una cadena de texto que incluya información adicional. De toda la información que se podría añadir al modelo se ha decidido mantener esta cadena simple, para que el modelo aprenda lo máximo posible del cuerpo de los comentarios. La estructura de esta cadena de texto es la siguiente: '*This is a comment about {**title**} in relation to {**topic**}.'* Como podemos ver la cadena de texto incluye el título de la propuesta a la que pertenece el comentario, esto le da información de contexto sobre la propuesta sin la necesidad de incluir el texto entero de la propuesta. Una alternativa a utilizar el título podría haber sido usar un modelo que resumiese la propuesta en los puntos importantes o en una frase breve, ya que se puede dar el caso de que el título de la propuesta no sea informativo y no recoja realmente la información que se quiere transmitir en la propuesta. Además, se añade la información del tema al que pertenece la propuesta, que como vimos en el análisis puede llegar a ser importante para determinar la postura del comentario. A los diferentes temas se les ha cambiado ligeramente la forma en la que estaban escritos (Tabla 4). El nuevo texto utilizado buscar reflejar en más detalle el contexto del tema utilizando lenguaje natural. Por ejemplo, el tema '*ValuesRights'* queda mejor expresado en lenguaje natural como '*values and rights*'. Esto puede mejorar el comportamiento del modelo ya que, por ejemplo, la palabra "*ValuesRights*" es posible que sea desconocida para el modelo y no sepa relacionarla con otros conceptos, pero '*values*' y '*rights'* por separado sí que serán conocidas por el modelo y le servirá para identificar mejor el contexto de los mensajes.

Tabla 4. Equivalencia entre el texto usado para identificar cada uno de los temas originales y el nuevo texto reescrito.

Temas originales	Temas reescritos
Migration	migration
GreenDeal	sustainable development
Health	health
Economy	economy

EUInTheWorld	europe in the world
ValuesRights	values and rights
Digital	digital
Democracy	democracy
Education	education
OtherIdeas	other topics

El conjunto de estás dos cadenas de texto será la entrada para los diferentes modelos de lenguaje natural empleados en este trabajo.

> *Características del ejemplo anterior: "this is a comment about Stop supporting corrupt governments in migrant countries in relation to migration."*

6.2. Traducción de los conjuntos de datos al inglés

Como se ha comentado en el análisis exploratorio de los datos, los comentarios se encuentran en diferentes idiomas. Para este tipo de datos se pueden utilizar modelos que estén preparados para lidiar con los diferentes idiomas. Otra posibilidad es traducir todos los datos a un mismo idioma y utilizar un modelo que este especializado solamente en ese idioma. Esto puede permitir al modelo aprender características más complejas del texto ya que no necesita tener que diferenciar entre los diferentes idiomas.

En este trabajo se han utilizado tanto modelos multilenguaje como modelos monolingües en inglés. Para poder utilizar estos últimos ha sido necesario traducir los conjuntos de datos al inglés. Se han elegidos modelos monolingües en inglés porque este es el idioma más utilizado en los modelos de lenguaje natural, ya que es el lenguaje más utilizado y existe una enorme producción de documentos que sirven para el entrenamiento.

Para la traducción de los comentarios en este trabajo se ha utilizado la librería de Python *deep-translator*[12] que proporciona una API de código libre, gratis de usar, y sencilla, para comunicarte con los principales servicios de traducción como *Google translate, DeepLTranslator* o *ChatGpt*.

Para este trabajo se ha utilizado la API de *Google translate* que proporciona resultados excelentes y el tiempo de ejecución es rápido. Para su uso, hay que indicar el idioma de entrada, en este caso se deja en automático ya que puede cambiar según el comentario, y el idioma de salida, para nuestro objetivo se selecciona el inglés. De este modo, se han traducido todos los comentarios de los diferentes conjuntos de datos utilizados en este trabajo.

6.3. *Data Augmentation* de los comentarios por traducción

Como se ha explicado anteriormente en el punto 3.3, el *data augmentation* consiste en aumentar el tamaño del conjunto de datos usando diferentes estrategias. En nuestro caso se ha realizado un *data augmentation* por traducción en el conjunto de datos CF_E-Dev. El proceso ha consistido en traducir cada uno de los comentarios a un idioma diferente y luego volver a traducirlo al idioma original. Utilizando este proceso se ha pasado de 1131 a 4524 comentarios, un aumento de x4.

En este conjunto de datos solo encontramos los idiomas: inglés, francés, español y alemán. La estrategia que se utilizó fue traducir cada comentario a cada uno de los otros tres idiomas y luego de vuelta al idioma original. Por tanto, para cada comentario tenemos 3 versiones nuevas.

En este caso se utilizó el modelo *'Helsinki-NLP/opus-mt-ine-ine'*[13] que se puede encontrar en la API de *HuggingFace*. Esta API es ligeramente más compleja de utilizar que la API de *deep-translator,* ya que al principio del texto a traducir hay

[12] https://deep-translator.readthedocs.io/en/latest/
[13] https://huggingface.co/Helsinki-NLP/opus-mt-ine-ine

que indicar el lenguaje de destino como un token *“>>id<<”* (id = identificador del lenguaje objetivo). El lenguaje de origen lo identifica automáticamente. Inicialmente se utilizó este modelo porque parecía una opción más adecuada para manejar las traducciones a idiomas diferentes al inglés. Sin embargo, después de algunas pruebas, las traducciones podrían haber sido igual de buenas utilizando la API de *deep-translator* comentada en el apartado anterior. Finalmente, como se tenían todos los comentarios ya generados utilizando la nueva API se decidió conservarlos y continuar con ellos ya que el conste computacional de la traducción era relativamente alto.

Aquí podemos observar un ejemplo de un comentario y las 3 versiones generadas. Vemos que los comentarios son diferentes, han cambiado ciertas palabras y en líneas generales los comentarios transmiten el mismo mensaje. Es cierto que se comete ciertos errores, por ejemplo, la parte final “a que intereses económicos obedecen” no está traducida correctamente y se pierde el verdadero significado de esa parte de la frase.

> *Original - Únicamente podemos incentivar constantemente a la creación de nuevos medios digitales, evitar que se les silencie en internet y obligar, a todos, a informar transparentemente a que intereses económicos obedecen.*
>
> *1 - Sólo podemos invocar constantemente la creación de nuevos medios digitales, evitar que se silencien en Internet y obligar a todos a informar claramente que los intereses económicos son obedientes.*
>
> *2 - Podemos simplemente impulsar constantemente la creación de nuevos medios digitales, evitar silencios en Internet y a todos para informar transparentemente de que los intereses económicos son obedientes.*
>
> *3 - Sólo podemos entrar constantemente en la creación de nuevos medios digitales, evitar que se descansen en Internet y*

informan transparentemente lo que obedezcan los intereses económicos.

6.4. *Label spreading* en el conjunto de datos CF_U

Se ha utilizado la técnica de *label spreading* para propagar las etiquetas del conjunto CF_E-Dev sobre el conjunto sin etiquetar CF_U. De este modo se ha pasado de 1400 comentarios etiquetados a más de 12000, un aumento cercano a un orden de magnitud.

La propagación de las etiquetas se ha realizado en dos pasos. Primero, se han generado los *embeddings* de los comentarios de los conjuntos de datos CF_U y CF_E-Dev. Para ello se ha usado el modelo *'paraphrase-multilingual-mpnet-base-v2*[14] que se puede encontrar en la API de *HuggingFace* (Reimers & Gurevych, 2019). Este modelo está entrenado en más de 50 idiomas, así que es adecuado para el tipo de datos que tenemos. La salida que produce es el comentario en formato de *embedding*.

En segundo lugar, se utiliza el modelo *LabelSpreading* de *scikit-learn*[15] para propagar las etiquetas de los embeddings de CF_E-Dev, a los *embeddings* de CF_U. La propagación consiste en un algoritmo semi-supervisado que asigna la misma etiqueta según la matriz de similaridad de los *embeddings*.

Tabla 5. Tabla con tres comentarios de ejemplos cuya etiqueta ha sido asignada mediante *label spreading*.

Propuesta	Comentario	Etiqueta
Ban the shark fin trade in Europe.	No deberíamos olvidarnos de la cruel matanza que se hace cada año por "tradición " en las islas Feroe, bajo	In favor

[14] https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[15] https://scikit-learn.org/stable/modules/semi_supervised.html#label-propagation

	protectorado danés. Este año 1428 delfines asesinados.	
No more immigration from non-european or non first word countries in relation to migration.	¡Entonces lo que necesitamos son mayores programas de integración para evitar que se formen eso guetos! El problema con la "falta de integración" y el "mal del multiculturalismo" viene por la falta de interés de los gobiernos de la Unión para financiar potentes programas de acojida e inserción en el tejido social del país.	Against
Reducing Imported Greenhouse Gas Emissions	La propuesta del comentario de Anton nonne no cumpliría con el derecho de competencia internacional, y está más focalizada en la economía que en el cambio climatico	Other

En la Tabla 5 vemos tres ejemplos de comentarios que han sido etiquetados mediante la técnica de *label spreading* comentada anteriormente. En estos tres casos podemos afirmar que la etiqueta ha sido asignada de manera correcta. En el primer caso, el escritor está a favor de la prohibición y da un motivo de porqué se debería llevar a cabo dicha prohibición. En el segundo caso, podemos afirmar que el comentario está en contra de la prohibición de los inmigrantes extracomunitarios, y explica cuáles serían las alternativas a esa prohibición. Por último, el tercer caso es el más difícil de juzgar porque la posición asignada es “neutral”. Sin embargo, en el comentario no aparece ningún motivo a favor o en contra de la propuesta, y realmente está criticando a otro comentario, así que se podría considerar que la clasificación correcta de este comentario es la neutral.

Hemos visto con estos ejemplos que esta aproximación puede llegar a ser muy útil para propagar la etiqueta de manera correcta entre los comentarios. Sin embargo, aquí hemos mostrados ejemplos en español, que es uno de los idiomas que encontramos en el conjunto de datos etiquetados. Posiblemente en el resto de los idiomas que no aparecen en el conjunto de datos etiquetado la propagación de las etiquetas es posible que no sea tan buena.

6.5. Descripción de los modelos utilizados

En total se han propuesto 10 modelos diferentes. Estos modelos se han entrenado como los diferentes conjuntos de datos preprocesados como se ha descrito anteriormente. La principal diferencia entre los modelos ha sido el número de lenguajes en el conjunto de entrada. Cuando todos los comentarios estaban traducidos al inglés se han usado modelos de tipo "**Roberta-base**", y cuando se han usado los conjuntos de datos multilingües se ha usado el modelo "**xlm-roberta-large**".

En general, no se han hecho ajuste de los hiperparámetros y se han usado en su mayoría los valores por defecto o se han ajustado para que el entrenamiento no sea demasiado lento, pero asegurando que el modelo está aprendiendo. Los hiperparámetros utilizados se describen en el **Anexo 1**.

Resumen de los modelos entrenados:

- **Modelo 1**. Entrenamiento de un modelo "**Roberta-base**"[16] con el conjunto de datos **CFE-D** (3 etiquetas) traducido al **inglés**.
- **Modelo 2**. Entrenamiento de un modelo "**xlm-roberta-large**"[17] con el conjunto de datos **CFE-D** (3 etiquetas).

La motivación de los Modelos 1 y 2 es el estudio de la influencia del conjunto de datos traducido al inglés. Con el Modelo 1, partimos de la hipótesis de que

[16] https://huggingface.co/roberta-base
[17] https://huggingface.co/xlm-roberta-large

el idioma original del comentario es irrelevante acerca de la postura de este. Por tanto, es preferible traducir los comentarios y utilizar un modelo especializado en un único idioma. Este modelo es capaz de aprender conceptos más complejos al no tener que dominar varios idiomas. Por otro lado, el Modelo 2, al ser un modelo multilingüe, si podrá detectar características específicas de los idiomas que puedan afectar a la detección de la postura y servirá para contrastar la hipótesis del Modelo 1.

- **Modelo 3**. Entrenamiento de un modelo "**Roberta-base**"[18] con el conjunto de datos **CFS** (2 etiquetas) traducido al **inglés.**
- **Modelo 4**. Entrenamiento de un modelo "**xlm-roberta-large**"[19] con el conjunto de datos **CFS** (2 etiquetas).

La hipótesis de partida para los Modelos 3 y 4 es similar a la de los Modelos 1 y 2, pero utilizando un conjunto de datos diferente. Aquí, se estudiará como afecta el uso de un conjunto de datos de partida con propiedades diferentes, un mayor número de casos y un número de etiquetas menor, y se comparará con las conclusiones de los Modelos 1 y 2.

- **Modelo 5**. Se ha entrenado un modelo "**bert-base-uncased**"[20] en **dos etapas** de *transfer learning*. En la primera se entrena con el conjunto de datos que tiene dos etiquetas en **inglés** (**CFS**), pero es más numeroso, y la segunda etapa con el conjunto de datos **CFE-D** que tiene las tres etiquetas traducido al **inglés.**
- **Modelo 6**. Se ha entrenado un modelo "**bert-base-multilingual-uncased**"[21] en **dos etapas** de *transfer learning*. El proceso es similar al Modelo 5 salvo que en este caso se han usado los conjuntos de datos en sus idiomas originales.

Los Modelos 5 y 6 sirven de nuevo para comparar entre ellos la influencia del uso de los conjuntos de datos traducidos. Además, comparando con los modelos

[18] https://huggingface.co/roberta-base
[19] https://huggingface.co/xlm-roberta-large
[20] https://huggingface.co/bert-base-uncased
[21] https://huggingface.co/bert-base-multilingual-uncased

anteriores, aquí se va a realizar un método de entrenamiento en dos etapas de *transfer learning*, lo cual va a permitir que los modelos aprendan tanto del conjunto CFS, como del CFE-D. Por tanto, se va a estudiar si la realización de esta rutina de entrenamiento más compleja es beneficiosa.

- **Modelo 7**. Se ha entrenado un modelo "**roberta-base**"[22] usando el conjunto de datos **CFU** (***Label spreading***) traducido en **inglés.**
- **Modelo 8**. Se ha entrenado un modelo " **xlm-roberta-large**"[23] usando el conjunto de datos **CFU** (***Label spreading***) sin traducir**.**

La motivación de los Modelos 7 y 8 es la aplicación de la técnica de *Label Spreading* para poder entrenar utilizando el conjunto de datos CFU. Se va a estudiar si la aplicación del *label spreading* mejora los resultados de los modelos anteriores. Además, se compararán los Modelos 7 y 8 entre ellos para estudiar la influencia del uso de conjuntos de datos traducidos, igual que en los pares de modelos anteriores.

- **Modelo 9**. Se ha entrenado un modelo " **xlm-roberta-large**"[24] usando el conjunto de datos **CFE-D** con ***data augmentation*** por traducción.

Con el Modelo 9 se plantea estudiar la Influencia del *data augmentation*. Los resultados de este modelo se compararán especialmente con el Modelo 2 que usa el mismo conjunto de datos, pero sin la aplicación de la *augmentation*.

- **Modelo *Ensemble***. Se ha entrenado un modelo **XGBoost**[25] (T. Chen & Guestrin, 2016) con las salidas de los **Modelos 1, 2, 3, 4, 5 y 6**. Junto con la información estructurada de los comentarios (votos positivos y negativos, *endorsements*…).
    - Los datos de entrenamiento en este modelo son las probabilidades asignadas a cada clase por los **modelos 1 al 6 al conjunto de entrenamiento CFE-D.** Los modelos 1, 2, 5 y 6 dan probabilidades

[22] https://huggingface.co/roberta-base
[23] https://huggingface.co/xlm-roberta-large
[24] https://huggingface.co/xlm-roberta-large
[25] https://xgboost.readthedocs.io/en/stable/

de las 3 clases ya que fueron entrenados con conjuntos de datos que tienen las 3 etiquetas. Mientras que los modelos 3 y 4 solo darán probabilidades para 2 etiquetas.

- Además, añade como valores numéricos:
  - Número de votos positivos
  - Número de votos negativos
  - *Endorsements* de la proposición.
  - "Último comentario en hilo" como una variable booleana codificada con 0 y 1.
  - 'Las variables lenguaje', 'lenguaje de la proposición' y 'tema' codificadas en variables categóricas con código en números enteros.

La motivación del ***Ensemble*** es la unificación en un único modelo de la información aprendida por los modelos de lenguaje natural anteriores junto con el resto de la información estructurada. Este modelo debería ser capaz de aprender de los puntos fuertes de los diferentes modelos y complementarlo con la información estructurada. Por tanto, se espera que este modelo sea capaz de mejorar los resultados del mejor de los modelos que lo conforman.

6.6. Métricas

Las siguientes métricas son las más comúnmente utilizadas en problemas de clasificación para evaluar el rendimiento de un modelo de aprendizaje automático:

- **Precisión** (***Precision***): La precisión es una métrica que mide la proporción de instancias positivas identificadas correctamente por el modelo en relación con todas las instancias que el modelo clasificó como positivas (tanto las verdaderas como las falsas). En otras palabras, la precisión se enfoca en cuántas de las predicciones positivas del modelo son realmente correctas.

Fórmula: precision = VP / (VP + FP)

- **Exhaustividad** (***Recall***): La exhaustividad, también conocida como sensibilidad o *recall*, mide la proporción de instancias positivas que el modelo ha identificado correctamente en relación con todas las instancias positivas reales. Es decir, la exhaustividad se enfoca en cuántas instancias positivas reales son capturadas correctamente por el modelo.

  Fórmula: recall = VP / (VP + FN)

- **Puntuación F1** (***F1-score***): La puntuación F1 es una medida que combina la precisión y la exhaustividad en un solo valor. Es útil cuando se busca un equilibrio entre estas dos métricas. La puntuación F1 es la media armónica de la precisión y la exhaustividad.

  Fórmula: F1-score = 2 * (precision * recall) / (precision + recall)

Estas tres métricas se pueden calcular por separado para cada clase, y son útiles para evaluar cómo funciona el modelo para cada clase por separado. Para evaluar el modelo en su conjunto encontramos otras métricas como:

- **Exactitud** (***accuracy***): La exactitud se define como la proporción de ejemplos clasificados correctamente en relación con el total de ejemplos. La exactitud proporciona una medida general del rendimiento del modelo al evaluar su capacidad para clasificar correctamente las muestras. Sin embargo, en presencia de desequilibrio de clases, la exactitud puede no ser una métrica representativa del rendimiento real del modelo.

  Fórmula: Accuracy = (VP + VN) / (P+N)

- **Macro-F1**: El macro-F1 es una variante del F1-score que se calcula como media aritmética del F1-score de cada clase. No tiene en cuenta el desequilibrio de clases y trata todas las clases por igual. Es útil cuando se desea evaluar el rendimiento del modelo de manera equitativa en todas

las clases. Es especialmente adecuado cuando el conjunto de datos está desequilibrado.

En la competición se describe que la métrica a tener en cuenta para la clasificación es la Macro-F1, por tanto, esta será la métrica que se intente maximizar en los modelos. De todas formas, se utilizarán el resto de métricas descritas para añadir información a los resultados obtenidos.

6.7. Evaluación de los modelos

6.7.1. Validación cruzada estratificada

La evaluación de los modelos se realiza por validación cruzada estratificada en **tres partes**. En cada una de las **tres partes** el modelo se entrena un número arbitrario de ***epochs***, para cada *epoch* se calcula la **macro-F1** y se selecciona como mejor modelo para cada una de las **tres partes** aquel que tiene una **macro-F1** mayor. Por tanto, se acaba obteniendo un modelo optimizado para cada una de las tres partes de la validación cruzada estratificada. Por último, se compara la puntuación **macro-F1** de los tres modelos y se selecciona como modelo definitivo aquel con una puntuación mayor.

6.7.2. Cálculo de las probabilidades sin sobreajustar

Para poder entrenar el **Modelo *Ensemble***, es necesario obtener las probabilidades asignadas a cada clase por cada uno de los modelos con componen el *Ensemble* para el conjunto de entrenamiento **CFE-D**.

Para cada uno de estos modelos, cuando el conjunto de entrenamiento es diferente al conjunto de entrenamiento **CFE-D**: Primero, se obtiene el modelo definitivo según el apartado anterior (**6.7.1**). Y luego, se calculan las probabilidades que el modelo asigna al conjunto **CFE-D.**

Cuando el conjunto de entrenamiento del modelo ha sido el conjunto **CFE-D**, hay que realizar un proceso especial para evitar sobreajuste y fuga de información.

Para ello, se usan los 3 modelos obtenidos en cada uno de las tres partes de la validación cruzada. Cada uno de ellos se usa para predecir la parte del conjunto asignada al test de la validación cruzada. De esta manera nos aseguramos que tenemos las probabilidades sin sobreajustar para cada una de las 3 partes en las que el conjunto de datos se divide al realizar la validación cruzada, evitando fuga de información.

6.8. *Baseline*

Para comprobar la importancia de las correlaciones entre las variables numéricas y categóricas, sin usar el texto de los comentarios ni de las propuestas, se ha entrenado un clasificador *random forest* (*baseline*) y se han comparado los resultados obtenidos con un modelo *dummy* que predice todo *A favor*. Se ha elegido que el modelo *dummy* predice todo *A favor* porque es la clase más numerosa y, por tanto, la exactitud será mayor.

Para el modelo *dummy* se obtiene una exactitud del 0.53 y un Macro-F1 de 0.23. Sin embargo, para el *baseline* se obtiene una exactitud de 0.76 y un Macro-F1 de 0.41. El modelo es capaz de mejorar significativamente tanto exactitud como la Macro-F1. Esto nos corrobora que la información que contienen las variables numéricas y categóricas es útil para la clasificación de los comentarios.

En la Imagen 8 podemos ver que las características más importantes para el modelo han sido los números de votos positivos y negativos.

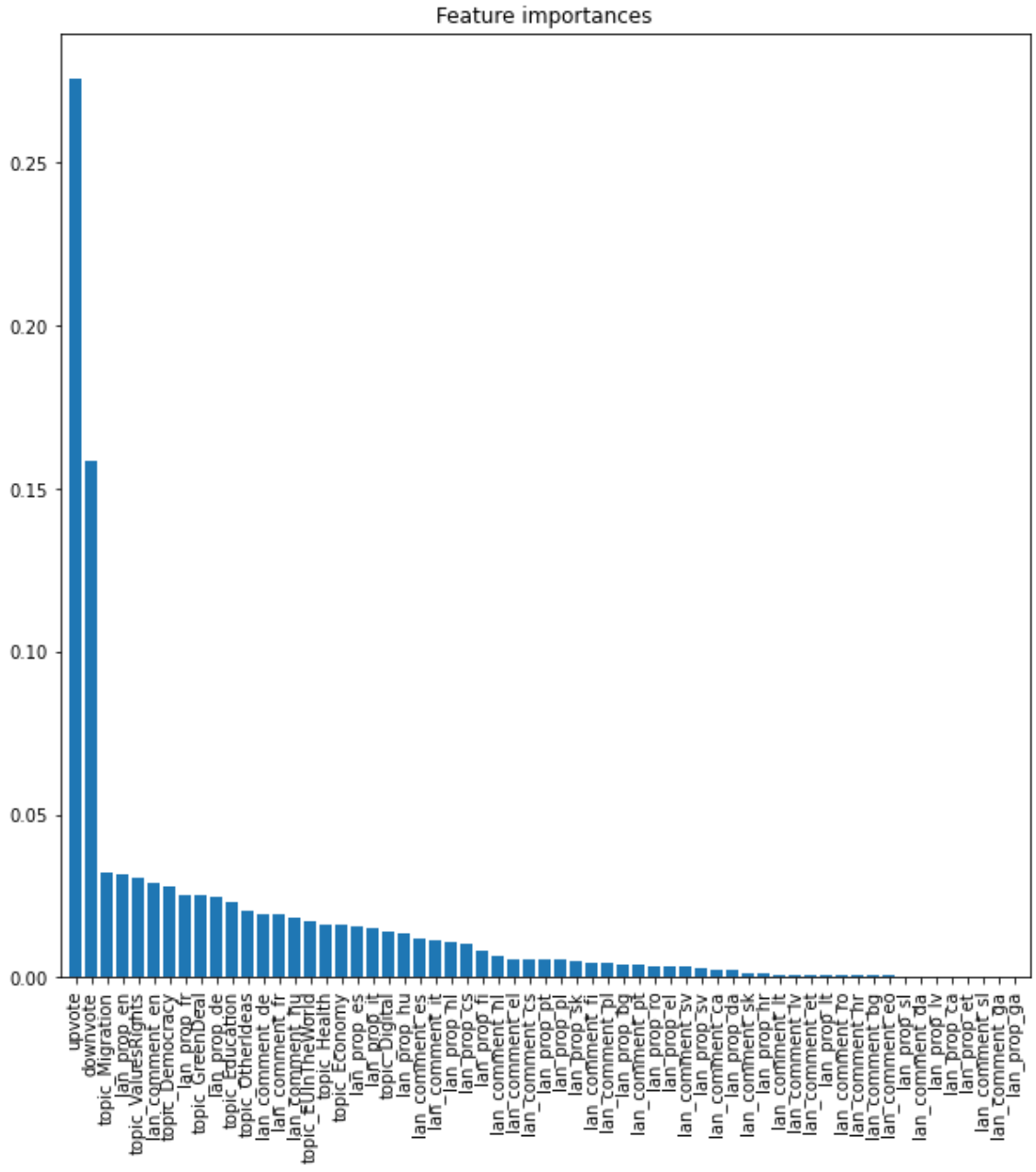

Imagen 8. Representación de la importancia de las características del modelo propuesto como *baseline*.

7. Análisis y Discusión de Resultados

En este capítulo, se presentan y analizan los resultados de la evaluación de los modelos desarrollados en este trabajo. Los modelos han sido evaluados utilizando las métricas estándar de clasificación descritas en el capítulo anterior. En la Tabla 6 encontramos un resumen de los resultados para las métricas principales. En el Anexo 2, se pueden encontrar en detalle los resultados al completo para cada modelo.

Tabla 6. Resumen de la evaluación de los diferentes modelos utilizados. El modelo "In favor" es un modelo de referencia que predice todo "*In Favor*". El modelo *Baseline* es un modelo de referencia que utiliza solo la información tabular descrito en "6.8. *Baseline* Se ha sombreado en azul el valor más alto para cada parámetro, y en naranja los casos en los que los modelos desarrollados superan al *Baseline*.

Modelo	Exactitud (accuracy)	Macro F1-score	"Against" F1-score	"In Favor" F1-score	"Other" F1-score
"In Favor"	0.53	0.23	0.00	0.69	0.00
Baseline	**0.76**	**0.41**	**0.37**	**0.86**	**0.02**
Modelo 1	0.64	0.54	0.29	0.73	0.60
Modelo 2	0.70	0.61	0.41	0.78	0.65
Modelo 3	0.55	0.39	0.42	0.74	0.00
Modelo 4	0.55	0.39	0.43	0.74	0.00
Modelo 5	0.66	0.60	0.47	0.74	0.60
Modelo 6	0.64	0.55	0.35	0.71	0.60
Modelo 7	0.68	0.60	0.44	0.75	0.60
Modelo 8	0.71	0.62	0.41	0.78	0.67
Modelo 9	0.66	0.58	0.43	0.75	0.56
Ensemble	**0.73**	**0.67**	**0.52**	**0.81**	**0.68**

7.1. Modelos de referencia: *Baseline* e '*In Favor*'

Como referencia para comparar la evaluación de los diferentes modelos se han añadido a la tabla los resultados del modelo ***Baseline*** (apartado **6.8**) e '***In Favor***'. El modelo '***In Favor***' es un modelo *Dummy* que predice todos los resultados a

'*In Favor'*. Se ha añadido este modelo ***'In Favor'*** porque es el propuesto por la organización de la competición como referencia.

Sin embargo, vemos que el modelo ***Baseline*** supera al modelo ***"In Favor"*** en todos los parámetros. Con respecto a la clasificación de las diferentes etiquetas, este modelo obtiene mejores puntuaciones F1 en los tres casos. Además, la Macro-F1 también es superior. Por tanto, para la evaluación de nuestros modelos compararemos preferentemente frente al modelo ***Baseline***.

7.2. Comparación frente al *Baseline* y desempeño del modelo *Ensemble*

En la Tabla 6 se ha marcado en azul los casos con una puntuación mayor, y en naranja los casos que superan al ***Baseline*** para cada parámetro. Podemos observar que, salvo la Exactitud, y la puntuación F1 de "In Favor", el resto de parámetros han sido superados por los modelos propuestos. Entre ellos destaca el modelo ***Ensemble*** que es el modelo que supera en mayor medida al ***Baseline*** en el resto de parámetros. Podemos resaltar el aumento considerable en la puntuación Macro-F1, pasando de 0.41 a 0.67. Con respecto al resto de modelos, solo los **modelos 3 y 4**, quedan por debajo del ***Baseline*** para la puntuación Macro-F1. Siendo estos dos modelos aquellos que se entrenaron con solo dos etiquetas, es comprensible que su desempeño general se vea perjudicado. Sin embargo, para el resto de modelos, la alta puntuación Macro-F1 es un buen indicativo de que puedan obtener unos buenos resultados en la competición.

Con respecto al resto de parámetros, el modelo ***Ensemble*** también mejora en gran medida la puntuación F1 para la clase "*Other*", pasando de 0.02 a 0.68. Aquí podemos concluir que el modelo ***Baseline*** no ha sido capaz de identificar la clase "*Other*" y se ha centrado principalmente en la clase "*In Favor*", explicando la alta puntación que muestra para esta clase, que no ha podido ser superada por ningún otro modelo. Por otro lado, la mejor clasificación que hacen, en

general, los modelos para las tres clases perjudican la exactitud, explicando que los modelos tengan valores de exactitud por debajo del ***Baseline***. De todos modos, el modelo ***Ensemble*** consigue prácticamente mantener la Exactitud del ***Baseline***, disminuyendo solo de 0.76 a 0.73.

Por otro lado, los modelos del **1** al **9** tienen problemas en superar valores de Macro-F1 de 0.60, y solo el modelo ***Ensemble*** consigue superarlo en gran medida llegando a 0.67. Esto podría ser debido a que, por particularidades intrínsecas de los conjuntos de datos o los métodos de entrenamiento utilizados, los modelos de Lenguaje Natural llegan a un límite alrededor de 0.60. Mientras que el modelo *Ensemble*, que incorpora también la información tabular, tiene acceso a más información y alcanza resultados mejores.

En definitiva, estos resultados muestran que el modelo ***Ensemble*** es capaz de mejorar los resultados de los modelos que lo conforma siendo este el que alcanza una Macro-F1 mayor.

7.3. Influencia de la complejidad del entrenamiento y el idioma de los conjuntos de datos

En este apartado vamos a comparar los modelos **1**, **2**, **3**, **4**, **5** y **6**, para estudiar la diferencia entre usar los conjuntos de datos originales frente a los traducidos al inglés, y la influencia del número de etapas y el número de etiquetas o clases en el entrenamiento. En la Tabla 1Tabla 7 se muestra de nuevo los resultados de estos modelos, donde se ha añadido también el tipo de entrenamiento y el idioma del conjunto de entrenamiento para facilitar el seguimiento de este apartado.

Tabla 7. Resumen de los resultados de los modelos 1, 2, 3, 4, 5 y 6. Se ha incluido el tipo de entrenamiento indicando si se ha entrenado con dos o tres etiquetas o si se ha entrenado en 2 etapas. Además, en el campo *Idioma* se especifica el idioma del conjunto de datos. Los modelos están ordenados de menor a mayor complejidad, aumentando hacia abajo el número de etiquetas y etapas en el entrenamiento.

Modelo	Tipo de entrenamiento	Idioma dataset	Exactitud (accuracy)	Macro F1-score	"Against" F1-score	"In Favor" F1-score	"Other" F1-score
Modelo 3	2 etiquetas	Inglés	0.55	0.39	0.42	0.74	0.00
Modelo 4	2 etiquetas	Multi	0.55	0.39	0.43	0.74	0.00
Modelo 1	3 etiquetas	Inglés	0.64	0.54	0.29	0.73	0.60
Modelo 2	3 etiquetas	Multi	0.70	0.61	0.41	0.78	0.65
Modelo 5	2 etapas (2 y 3 etiquetas)	Inglés	0.66	0.60	0.47	0.74	0.60
Modelo 6	2 etapas (2 y 3 etiquetas)	Multi	0.64	0.55	0.35	0.71	0.60

Empezando por los modelos **3** y **4**, vemos que obtienen puntuaciones prácticamente idénticas, existiendo solo una diferencia de 0.01 para la puntuación F1 para la clase *Against*. Podemos concluir que para el entrenamiento usando solo dos etiquetas, el idioma del conjunto de datos no es importante. Esto puede ser debido a que al faltar datos de entrenamiento sobre una de las clases el nivel de error que esto introduce es mayor que las diferencias que podrían suponer el uso de conjuntos de datos traducidos.

Por otro lado, los modelos **1** y **2**, que han sido entrenados con conjuntos de datos que contienen información sobre las tres etiquetas, son capaces de mejorar las puntuaciones de los modelos **3** y **4.** Además, el modelo **2** es sustancialmente superior al modelo **1** en todos los parámetros alcanzando una macro-F1 de 0.61. Estos resultados respaldan que introduciendo la información de la clase que faltaba, no solo somos capaces de mejorar sustancialmente los resultados, sino que empezamos a ver diferencias entre los modelos dependiendo del idioma del conjunto de datos utilizados. En este caso el modelo **2** fue entrenado usando el conjunto de datos sin traducir incluyendo todos los idiomas diferentes.

Sin embargo, para los modelos **5** y **6**, que fueron entrenados en dos etapas, primero con el conjunto de datos con dos etiquetas y luego con el conjunto de datos de tres etiquetas, vemos que es el modelo **5**, que fue entrenado con los conjuntos de datos traducidos al inglés, el que obtiene la mayor puntuación.

Comparando los modelos **2** y **5**, obtenemos puntuaciones muy similares para ambos, especialmente el modelo **2** presenta una macro-F1 mayor solo por 0.01. Sin embargo, si comparamos los modelos a nivel del lenguaje de los conjuntos de datos, vemos que al pasar del modelo **2** al **6**, que usan los idiomas originales, los parámetros empeoran, pero al pasar del modelo **1** al **5,** que usan los datos traducidos, los parámetros mejoran. No está claro el origen de este efecto, pero el aumento de la complejidad en el entrenamiento solo es beneficioso en el caso del usar los comentarios traducidos.

Por tanto, la elección del modelo y aproximación correcta puede ser muy dependiente de cada caso y del conjunto de datos que se posea. Sin duda, el estudio de diferentes aproximaciones es muy recomendable para encontrar los mejores resultados.

7.4. Influencia del *Label Spreading* en el rendimiento de los modelos

Los modelos **7** y **8** fueron entrenados usando un conjunto de datos aumentado por *label spreading*. Los modelos con los que podemos comparar de forma más directa son el **1** y el **2**, ya que usan el mismo conjunto de datos, pero sin el aumento en comentarios etiquetados por el *label spreading*. En la Tabla 8 se han incluido los resultados de estos modelos para compararlos con más facilidad.

Tabla 8. Comparación de los resultados de los modelos entrenados sin *label spreading* (1 y 2) y los modelos entrenados utilizando el conjunto de datos expandido usando *label spreading*.

Modelo	Label spreading	Idioma dataset	Exactitud (accuracy)	Macro F1-score	"Against" F1-score	"In Favor" F1-score	"Other" F1-score
Modelo 1	No	Inglés	0.64	0.54	0.29	0.73	0.60
Modelo 2	No	Multi	0.70	0.61	0.41	0.78	0.65
Modelo 7	Sí	Inglés	0.68	0.60	0.44	0.75	0.60
Modelo 8	Sí	Multi	0.71	0.62	0.41	0.78	0.67

Si comparamos los modelos 2 y 8, que son los modelos que utilizan los comentarios sin traducir al inglés, vemos que se obtienen puntuaciones muy similares en todos los parámetros. Solo se obtiene aumentos del orden de 0.01 para el modelo 8, que usa *label spreading*. Por tanto, aunque el uso de *label spreading* mejora los parámetros, este aumento es muy pequeño y podemos concluir que, el aumento del conjunto de datos no aporta en gran medida información relevante al modelo, obteniéndose resultados muy similares.

Por otro lado, los modelos **1** y **7** que usan los conjuntos de datos traducidos al inglés, vemos que si hay una mejora significativa en los parámetros al implementar el *label spreading*. La puntuación macro-F1 llega a 0.60 en el caso del modelo **7**, muy similar a la de los modelos **2** y **8**. Vemos que, los posibles defectos que tenía el usar el conjunto de datos traducido se disminuyen al aumentar el número de comentarios. En estos resultados podemos volver a identificar que los modelos parecen alcanzar cierto limite alrededor de una puntuación de macro-F1 de 0.60.

7.5. Influencia del *Data augmentation* en los resultados del Modelo 9

El conjunto de datos utilizado para el entrenamiento del modelo **9** fue aumentado por la técnica de *data augmentation* por traducción descrita en el apartado 6.3.

El conjunto de datos resultante sigue siendo multilingüe y mantiene la misma distribución de idiomas que el conjunto original. Los resultados del modelo **9** se van a comparar con el modelo **2** que fue entrenado con el conjunto de datos original sin *data augmentation*. En la Tabla 9 encontramos los resultados de estos dos modelos.

Tabla 9. Resumen de los resultados de los modelos 2 y 9, donde se compara la influencia de la implementación del *data augmentation* por traducción en el conjunto de entrenamiento.

Modelo	*Data augmentation*	Exactitud (accuracy)	Macro F1-score	"Against" F1-score	"In Favor" F1-score	"Other" F1-score
Modelo 2	No	0.70	0.61	0.41	0.78	0.65
Modelo 9	Sí	0.66	0.58	0.43	0.75	0.56

Teniendo en cuenta que el conjunto de datos utilizado para entrenar el modelo **9** incorpora toda la información que tiene el conjunto de datos utilizado para entrenar el modelo **2**, se esperaría que el modelo **9** obtuviese como mínimo los mismos valores para los diferentes parámetros que el modelo **2**. Sin embargo, observamos que en general se obtienen valores más bajos para todos parámetros, menos para la puntuación F1 de la clase *Against*, comparando con el modelo **2**.

En definitiva, los datos augmentados parecen confundir al modelo **9** provocando el empeoramiento de los resultados. Una causa podría ser mala calidad en la traducción de ida y vuelta de los comentarios perdiéndose en el proceso la postura del comentario original.

8. Resultados de la participación en la competición

Como parte final del trabajo, los diferentes modelos descritos fueron utilizados para participar en la competición. Se enviaron un total de 6 ejecuciones diferentes para probar los distintos enfoques en la competición.

Para la preparación de las ejecuciones se realizó la predicción para el conjunto de test con el modelo correspondiente para cada una de las ejecuciones. Estos resultados se subieron a la plataforma de evaluación de la tarea, y unos días después hicieron pública las evaluaciones que comentamos más adelante. Es importante tener en cuenta que las etiquetas verdaderas del conjunto de test nunca se han hecho públicas y por tanto, no se han podido hacer comprobaciones u otros experimentos en local con este conjunto de datos.

Los resultados descritos aquí han sido publicados como parte de la competición y de la iniciativa *Touché*. En el momento en el que se escribe este documento, están disponibles las *working notes* que resumen toda la actividad de la iniciativa Touché en 2023.[26] Entre los diferentes preceedings se encuentra nuestra publicación describiendo nuestros resultados (Avila et al., 2023).[27]

8.1. Descripción de las ejecuciones enviadas

Se han enviado un total de 6 ejecuciones estructuradas del siguiente modo:

- **Run 1**. **Modelo *Ensemble***
  Como primera ejecución se utilizó el Modelo ***Ensemble***, ya que este fue el que obtuvo los mejores resultados en las evaluaciones realizadas en local.
- **Run 2**. Salida del **Modelo 7**
  El modelo **7** se entrenó con el conjunto de datos expandido mediante *label spreading*, con los comentarios traducidos al inglés. La motivación de

[26] https://www.dei.unipd.it/~faggioli/temp/CLEF2023-proceedings/
[27] https://www.dei.unipd.it/~faggioli/temp/CLEF2023-proceedings/paper-260.pdf

utilizar este modelo para la ejecución es comprobar si al haber sido entrenado utilizando un conjunto de datos mayor le permitiría generalizar mejor los casos desconocidos que el resto de modelos.

- **Run 3**. Salida del **Modelo 8**

  El caso del modelo **8** es similar al anterior, pero este fue entrenado con los comentarios sin traducir. En esta ejecución, queríamos estudiar la influencia del idioma en la propagación de etiquetas. En la evaluación en local el modelo **8** fue ligeramente superior al **7**.

- **Run 4**. Salida del **Modelo 9**

  A pesar de que en las evaluaciones en local el modelo **9** no estuvo entre los mejores, con este run se pretende comprobar si el uso del *data augmentation* en los datos de entrenamiento podría aumentar la capacidad de generalización del modelo con datos desconocidos y superar a los otros modelos.

- **Run 5**. Salida del **Modelo 1**

  Se ha elegido al modelo **1** para esta ejecución porque sirve de referencia frente a las ejecuciones anteriores que utilizan el *label spreading* o el *data augmentation*.

- **Run 6**. Salida del **Modelo 5**

  Como última ejecución se eligió el modelo 5 para probar el efecto del entrenamiento en dos etapas.

8.2. Análisis de los resultados

En la Tabla 10 mostramos los resultados de nuestras ejecuciones y el punto de referencia propuesto por los organizadores, junto con los resultados del único otro participante que también ha publicado sus resultados (*queen-of-swords*). Los resultados están ordenados según la medida oficial propuesta por la competición, la puntuación macro-F1.

Tabla 10. Resultados de las ejecuciones junto a la referencia (*touche23-baseline*) propuesta por los organizadores. Las ejecuciones *queen-of-swords* 1 y 2 se corresponden a los resultados de otro participante.

Ejecuciones	Modelo	Exactitud (accuracy)	Macro F1-score
queen-of-swords-1		0.605	0.417
Run 6	Modelo 5	0.551	0.35
Run 4	Modelo 9	0.537	0.329
queen-of-swords-2		0.616	0.324
Run 1	Ensemble	0.524	0.323
Run 5	Modelo 1	0.463	0.27
Run 2	Modelo 7	0.461	0.239
touche23-baseline		0.552	0.237
Run 3	Modelo 8	0.414	0.216

La referencia propuesta por la organización (*touche23-baseline*) es un modelo que predice todo '*In Favor'*. Este modelo obtiene una exactitud de 0.55, lo que quiere decir que más de la mitad de los comentarios son '*In Favor*' y por tanto el conjunto de test también está desbalanceado. La puntuación macro-F1 del modelo de referencia es baja (0.24), pero tanto la exactitud como la macro-F1 son muy similar a las que obtiene el modelo de referencia en el conjunto de entrenamiento (Exactitud:0.53, Macro-F1: 0.23).

Con respecto a las ejecuciones enviadas, podemos observar que todas, excepto la ejecución 3, superan la referencia. Sin embargo, todos los resultados están por debajo de 0.4, y comparando con las evaluaciones realizadas en local, estos resultados quedan muy por debajo del 0.67 obtenido por el modelo *Ensemble*. Por lo tanto, creemos que nuestros modelos no han sido capaces de generalizar correctamente los datos de entrenamiento. Una posible causa del peor desempeño de los modelos en el test es que en el conjunto de entrenamiento

principal (CF_E-D) solo tiene 4 idiomas diferentes, y sin embargo el conjunto de test está constituido por 6 idiomas, de los cuales 3 de ellos no aparecen en el entrenamiento. Esto no debería ser un problema para los modelos que usan los datos traducidos al inglés, pero podría afectar a los modelos que usan los comentarios sin traducir.

Los mejores resultados se obtienen con la ejecución 6, lo que demuestra la utilidad del entrenamiento en dos etapas, permitiendo la incorporación de datos adicionales que utilizan un número diferente de etiquetas. Además, en segundo lugar, encontramos la ejecución 4, demostrando la importancia de incluir datos de entrenamiento adicionales, obtenidos en esta ejecución mediante *data augmentation*. Para la ejecución 1, que utiliza el modelo *Ensemble*, se obtienen resultados similares. Sin embargo, es interesante que el *Ensemble* no sea capaz de superar a las otras ejecuciones (4 y 6), quedando un poco por debajo. Sobre todo, teniendo en cuenta que el modelo 5, que constituye la ejecución 6, también forma parte del *Ensemble*. Estos resultados parecen indicar que con el modelo *Ensemble* se sobreajustó a los datos de entrenamiento. Todas estas ejecuciones superaron a la ejecución 5, que se considera nuestra referencia.

Sin embargo, los resultados utilizando *label spreading* no fueron tan exitosos, como podemos observar en los resultados de la ejecución 3, la única ejecución peor que la referencia, y la ejecución 2. Ambas ejecuciones tuvieron una puntuación inferior a la ejecución 5, que utiliza el mismo conjunto de entrenamiento, pero sin expandir. Por lo tanto, es necesario investigar más a fondo cómo aprovechar adecuadamente los datos no etiquetados para que realmente aporten información a los modelos.

8.3. Análisis de los resultados del resto de participantes

En la publicación resumen donde se referencias los diferentes trabajos de los equipos que han participado en las distintas tareas,[28] solo aparecen resultados

[28] https://www.dei.unipd.it/~faggioli/temp/CLEF2023-proceedings/

de un equipo más. Este equipo, denominado *queen-of-swords,* realizó dos ejecuciones para esta tarea, cuyos resultados se han incluido en la Tabla 10 (Schaefer, 2023).[29]

Brevemente, la estrategia usada por este equipo fue entrenar un modelo BERT[30] con el conjunto de datos con tres etiquetas (CF_E-D) y utilizar este modelo para etiquetar el conjunto de datos sin etiquetar (CF_U). Por tanto, aquí están utilizado *label spreading* de manera similar a la desarrollada en este trabajo, pero usando un modelo BERT para expandir las etiquetas. Finalmente, entrenaron un nuevo modelo BERT utilizando los dos conjuntos de datos. Es importante destacar que los conjuntos de datos fueron inicialmente traducidos al inglés.

Esta estrategia les permitió alcanzar resultados muy buenos en las evaluaciones que realizaron en local, alcanzando una puntuación de 0.886 para la Macro-F1. Sin embargo, en las ejecuciones enviadas con el conjunto de test obtienen unas puntuaciones de 0.417 y 0.324, muy por debajo de sus evaluaciones en local. Por tanto, los resultados parecen indicar que sobreajustaron el modelo. Según su publicación, los hiperparámetros del modelo fueron ajustados usando un conjunto reducido de comentarios, entrenando con este conjunto reducido daría lugar al modelo utilizado para su primera ejecución (Macro-F1 de 0.417). Para la segunda ejecución (Macro-F1 de 0.324) entrenaron con todos los comentarios, pero mantuvieron los mismos hiperparámetros. A pesar de entrenar con un número mayor de comentarios, el no haber optimizado los hiperparámetros perjudico los resultados de esta ejecución.

Si comparamos con nuestros resultados, vemos que dos de nuestras ejecuciones (la 6 y la 4) quedan por encima de su segunda ejecución. Sin embargo, su primera ejecución queda bastante por encima, con 0.06 de puntuación por encima de nuestra mejor ejecución (Tabla 10). Comparando con nuestras ejecuciones que utilizan la técnica de *label spreading* (runs 2 y 3), su aproximación ofrece mucho mejores resultados.

[29] https://www.dei.unipd.it/~faggioli/temp/CLEF2023-proceedings/paper-265.pdf
[30] https://huggingface.co/bert-base-uncased

Como conclusión, vemos que todos los participantes hemos sufrido de grandes diferencias entre los resultados obtenidos en local y las ejecuciones en el conjunto de test, posiblemente indicando que hemos sufrido de sobreajuste. Por otro lado, podemos identificar diferentes estrategias que han dado lugar a buenos resultados, como el *label spreading* utilizado por *queen-of-swords*, o el entrenamiento en 2 etapas y uso de *data augmentation* por traducción utilizados en nuestras ejecuciones 6 y 4. Un posible trabajo futuro podría ser la combinación de estas tres técnicas para intentar superar los resultados obtenidos.

9. Conclusiones

En el presente TFM se han puesto en práctica los conocimientos adquiridos a lo largo del Master de Ingeniería y Ciencia de Datos. Dicho trabajo ha permitido explorar las capacidades para la clasificación de postura en textos de forma automática. La detección de postura en el procesamiento de lenguaje natural es una tarea fundamental con aplicaciones significativas en diversos campos, desde la gestión de la reputación en línea hasta la toma de decisiones políticas y públicas. A medida que el procesamiento de lenguaje natural avanza, se enfrenta al desafío de abordar la diversidad lingüística y contextual, lo que ha llevado al desarrollo de tareas como la evaluación *Intra-Multilingual Multi-Target Stance Classification 2023*.

Este trabajo ha demostrado la utilidad de diversas técnicas de aprendizaje automático y aprendizaje profundo para mejorar la clasificación de postura en un entorno multilingüe. El uso de estrategias como el *transfer learning*, *data augmentation* y *label spreading* ha permitido obtener resultados que superen a los modelos de referencia establecidos por la tarea. En total se han entrenado 10 modelos diferentes que combinan esas tres estrategias junto con la traducción de los conjuntos de datos al inglés. Entre estos modelos, se desarrolló un modelo *Ensemble* constituido por la combinación de la salida de un grupo de los modelos. El modelo *Ensemble* es una estrategia interesante porque permitía unificar el conocimiento del resto de los modelos e introducir información extra tabulada. En el proceso de evaluación interna, el modelo *Ensemble* fue el que alcanzo los mejores resultados con una Macro-F1 de 0.67. En la competición, la puntuación Macro-F1 más alta alcanzada fue de 0.35 con un modelo basado en *transfer learning* y entrenamiento en dos etapas. Los resultados alcanzados en el trabajo están en línea con los obtenidos por los otros participantes, llegando a superarse la puntuación que obtienen en uno de los casos. En panel de clasificación final, nuestra ejecución logró quedar en segundo lugar.

La combinación de los resultados de las ejecuciones realizadas en este trabajo junto con las realizadas por el resto de participantes podrían dar lugar a

estrategias que combinen los mejores procedimientos para obtener mejoras en los resultados.

Como resultado final, la participación desarrollada en la tarea *Intra-Multilingual Multi-Target Stance Classification* ha quedado publicada (Avila et al., 2023) como parte de la conferencia CLEF 2023.[31]

[31] https://clef2023.clef-initiative.eu/

Bibliografía

Anand, P., Walker, M., Abbott, R., Fox Tree, J. E., Bowmani, R., & Minor, M. (2011). Cats Rule and Dogs Drool!: Classifying Stance in Online Debate. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, 1-9. https://aclanthology.org/W11-1701

Avila, J., Rodrigo, A., & Centeno, R. (2023). Silver Surfer team at Touché task 4: Testing data augmentation and label propagation for multilingual stance detection. *Working Notes of CLEF 2023*. CEUR Workshop Proceedings, CEUR-WS.org, 2023.

Baly, R., Mohtarami, M., Glass, J., Màrquez, L., Moschitti, A., & Nakov, P. (2018). Integrating Stance Detection and Fact Checking in a Unified Corpus. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 21-27. https://doi.org/10.18653/v1/N18-2004

Barriere, V., & Balahur, A. (2020). Improving Sentiment Analysis over non-English Tweets using Multilingual Transformers and Automatic Translation for Data-Augmentation. *Proceedings of the 28th International Conference on Computational Linguistics*, 266-271. https://doi.org/10.18653/v1/2020.coling-main.23

Barriere, V., Balahur, A., & Ravenet, B. (2022). Debating Europe: A Multilingual Multi-Target Stance Classification Dataset of Online Debates. *Proceedings*

of the LREC 2022 workshop on Natural Language Processing for Political Sciences, 16-21. https://aclanthology.org/2022.politicalnlp-1.3

Barriere, V., Jacquet, G. G., & Hemamou, L. (2022). CoFE: A New Dataset of Intra-Multilingual Multi-target Stance Classification from an Online European Participatory Democracy Platform. *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 418-422. https://aclanthology.org/2022.aacl-short.52

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.48550/ARXIV.1603.02754

Chen, X., & Zhao, M. (2022). A Stable Community Detection Approach for Large-Scale Complex Networks Based on Improved Label Propagation Algorithm. En D.-S. Huang, K.-H. Jo, J. Jing, P. Premaratne, V. Bevilacqua, & A. Hussain (Eds.), *Intelligent Computing Methodologies* (Vol. 13395, pp. 288-303). Springer International Publishing. https://doi.org/10.1007/978-3-031-13832-4_25

Cignarella, A. T., Lai, M., Bosco, C., Patti, V., & Rosso, P. (2020). SardiStance @ EVALITA2020: Overview of the Task on Stance Detection in Italian Tweets. En V. Basile, D. Croce, M. Maro, & L. C. Passaro (Eds.), *EVALITA Evaluation of NLP and Speech Tools for Italian—December 17th, 2020* (pp. 177-186). Accademia University Press. https://doi.org/10.4000/books.aaccademia.7084

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). *Unsupervised Cross-lingual Representation Learning at Scale*. https://doi.org/10.48550/ARXIV.1911.02116

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://doi.org/10.48550/ARXIV.1810.04805

Dey, K., Shrivastava, R., & Kaushik, S. (2018). *Topical Stance Detection for Twitter: A Two-Phase LSTM Model Using Attention*. https://doi.org/10.48550/ARXIV.1801.03032

Evrard, M., Uro, R., Hervé, N., & Mazoyer, B. (2020). French Tweet Corpus for Automatic Stance Detection. *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 6317-6322. https://aclanthology.org/2020.lrec-1.775

Faulkner, A. (2014). Automated classification of stance in student essays: An approach using stance target information and the wikipedia link-based measure. *Proceedings of the 27th International Florida Artificial Intelligence Research Society Conference, FLAIRS 2014*, 174-179.

Gorrell, G., Kochkina, E., Liakata, M., Aker, A., Zubiaga, A., Bontcheva, K., & Derczynski, L. (2019). SemEval-2019 Task 7: RumourEval, Determining Rumour Veracity and Support for Rumours. *Proceedings of the 13th International Workshop on Semantic Evaluation*, 845-854. https://doi.org/10.18653/v1/S19-2147

Hardalov, M., Arora, A., Nakov, P., & Augenstein, I. (2021a). *A Survey on Stance Detection for Mis- and Disinformation Identification*. https://doi.org/10.48550/ARXIV.2103.00242

Hardalov, M., Arora, A., Nakov, P., & Augenstein, I. (2021b). Cross-Domain Label-Adaptive Stance Detection. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 9011-9028. https://doi.org/10.18653/v1/2021.emnlp-main.710

Hasan, K. S., & Ng, V. (2013). Stance Classification of Ideological Debates: Data, Models, Features, and Constraints. *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, 1348-1356. https://aclanthology.org/I13-1191

Hercig, T., Krejzl, P., Hourová, B., Steinberger, J., & Lenc, L. (2017). Detecting Stance in Czech News Commentaries. En J. Hlaváčová (Ed.), *Proceedings of the 17th ITAT: Slovenskočeský NLP workshop (SloNLP 2017)* (Vol. 1885, pp. 176-180). CreateSpace Independent Publishing Platform.

Joseph, K., Shugars, S., Gallagher, R., Green, J., Quintana Mathé, A., An, Z., & Lazer, D. (2021). (Mis)alignment Between Stance Expressed in Social Media Data and Public Opinion Surveys. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 312-324. https://doi.org/10.18653/v1/2021.emnlp-main.27

Küçük, D., & Can, F. (2018). *Stance Detection on Tweets: An SVM-based Approach*. https://doi.org/10.48550/ARXIV.1803.08910

Küçük, D., & Can, F. (2019). *A Tweet Dataset Annotated for Named Entity Recognition and Stance Detection*. https://doi.org/10.48550/ARXIV.1901.04787

Küçük, D., & Can, F. (2021a). Stance Detection: A Survey. *ACM Computing Surveys*, *53*(1), 1-37. https://doi.org/10.1145/3369026

Küçük, D., & Can, F. (2021b). Stance Detection: Concepts, Approaches, Resources, and Outstanding Issues. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2673-2676. https://doi.org/10.1145/3404835.3462815

Lai, M., Cignarella, A. T., Farías, D. I. H., Bosco, C., Patti, V., & Rosso, P. (2020). Multilingual stance detection in social media political debates. *Computer Speech & Language*, *63*, 101075. https://doi.org/10.1016/j.csl.2020.101075

Lai, M., Patti, V., Ruffo, G., & Rosso, P. (2018). Stance Evolution and Twitter Interactions in an Italian Political Debate. En M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, & F. Meziane (Eds.), *Natural Language Processing and Information Systems* (Vol. 10859, pp. 15-27). Springer International Publishing. https://doi.org/10.1007/978-3-319-91947-8_2

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://doi.org/10.48550/ARXIV.1907.11692

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016a). A Dataset for Detecting Stance in Tweets. *Proceedings of the Tenth*

International Conference on Language Resources and Evaluation (LREC'16), 3945-3952. https://aclanthology.org/L16-1623

Mohammad, S., Kiritchenko, S., Sobhani, P., Zhu, X., & Cherry, C. (2016b). SemEval-2016 Task 6: Detecting Stance in Tweets. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 31-41. https://doi.org/10.18653/v1/S16-1003

Patel, H., & Verma, J. P. (2023). Community Detection Using Label Propagation Algorithm with Random Walk Approach. En R. Dhavse, V. Kumar, & S. Monteleone (Eds.), *Emerging Technology Trends in Electronics, Communication and Networking* (Vol. 952, pp. 307-320). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-6737-5_25

Rajadesingan, A., & Liu, H. (2014). *Identifying Users with Opposing Opinions in Twitter Debates*. https://doi.org/10.48550/ARXIV.1402.7143

Reimers, N., & Gurevych, I. (2019, noviembre). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. http://arxiv.org/abs/1908.10084

Schaefer, K. (2023). Queen of Swords at Touché 2023: Intra-multilingual multi-target stance classification using BERT,. *Working Notes of CLEF 2023*. CEUR Workshop Proceedings, CEUR-WS.org, 2023.

Slovikovskaya, V. (2019). *Transfer Learning from Transformers to Fake News Challenge Stance Detection (FNC-1) Task*. https://doi.org/10.48550/ARXIV.1910.14353

Sobhani, P., Inkpen, D., & Zhu, X. (2017). A Dataset for Multi-Target Stance Detection. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 551-557. https://aclanthology.org/E17-2088

Somasundaran, S., & Wiebe, J. (2009). Recognizing Stances in Online Debates. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1*, 226-234.

Somasundaran, S., & Wiebe, J. (2010). Recognizing Stances in Ideological On-Line Debates. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 116-124. https://aclanthology.org/W10-0214

Sun, Q., Wang, Z., Li, S., Zhu, Q., & Zhou, G. (2019). Stance detection via sentiment information and neural network model. *Frontiers of Computer Science*, *13*(1), 127-138. https://doi.org/10.1007/s11704-018-7150-9

Taulé, M., Martí, M. A., Pardo, F. M. R., Rosso, P., Bosco, C., & Patti, V. (2017). Overview of the Task on Stance and Gender Detection in Tweets on Catalan Independence. *IberEval@SEPLN*. https://api.semanticscholar.org/CorpusID:958864

Taulé, M., Pardo, F. M. R., Martí, M. A., & Rosso, P. (2018). Overview of the Task on Multimodal Stance Detection in Tweets on Catalan #1Oct Referendum. *IberEval@SEPLN*. https://api.semanticscholar.org/CorpusID:51941439

Thomas, M., Pang, B., & Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, 327-335. https://aclanthology.org/W06-1639

Vamvas, J., & Sennrich, R. (2020). *X-Stance: A Multilingual Multi-Target Dataset for Stance Detection*. https://doi.org/10.48550/ARXIV.2003.08385

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. https://doi.org/10.48550/ARXIV.1706.03762

Vijayaraghavan, P., Sysoev, I., Vosoughi, S., & Roy, D. (2016). DeepStance at SemEval-2016 Task 6: Detecting Stance in Tweets Using Character and Word-Level CNNs. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 413-419. https://doi.org/10.18653/v1/S16-1067

Vychegzhanin, S. V., & Kotelnikov, E. V. (2019). Stance Detection Based on Ensembles of Classifiers. *Programming and Computer Software*, *45*(5), 228-240. https://doi.org/10.1134/S0361768819050074

Wei, W., Zhang, X., Liu, X., Chen, W., & Wang, T. (2016). pkudblab at SemEval-2016 Task 6: A Specific Convolutional Neural Network System for Effective Stance Detection. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 384-388. https://doi.org/10.18653/v1/S16-1062

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., … Rush, A. M. (2019). *HuggingFace's Transformers: State-of-the-art Natural Language Processing*. https://doi.org/10.48550/ARXIV.1910.03771

Xu, R., Zhou, Y., Wu, D., Gui, L., Du, J., & Xue, Y. (2016). Overview of NLPCC Shared Task 4: Stance Detection in Chinese Microblogs. En C.-Y. Lin, N. Xue, D. Zhao, X. Huang, & Y. Feng (Eds.), *Natural Language Understanding and Intelligent Applications* (Vol. 10102, pp. 907-916). Springer International Publishing. https://doi.org/10.1007/978-3-319-50496-4_85

Zarrella, G., & Marsh, A. (2016). MITRE at SemEval-2016 Task 6: Transfer Learning for Stance Detection. *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 458-463. https://doi.org/10.18653/v1/S16-1074

Zhou, Y., Cristea, A. I., & Shi, L. (2017). Connecting Targets to Tweets: Semantic Attention-Based Model for Target-Specific Stance Detection. En A. Bouguettaya, Y. Gao, A. Klimenko, L. Chen, X. Zhang, F. Dzerzhinskiy, W. Jia, S. V. Klimenko, & Q. Li (Eds.), *Web Information Systems Engineering – WISE 2017* (Vol. 10569, pp. 18-32). Springer International Publishing. https://doi.org/10.1007/978-3-319-68783-4_2

Zotova, E., Agerri, R., & Rigau, G. (2021). Semi-automatic generation of multilingual datasets for stance detection in Twitter. *Expert Systems with Applications*, *170*, 114547. https://doi.org/10.1016/j.eswa.2020.114547

Lista de imágenes y tablas

Imágenes

Tablas

Anexo 1. Hiperparámetros

Modelo 1:

- Batch size: 8
- Leaning rate: 6 x 10^{-6}
- Weight decay: 0.001
- Epochs: 6

Modelo 2:

- Batch size: 2 (con acumulación de 4, equivalente a 2x4=8 de batch size)
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001
- Epochs: 8

Modelo 3:

- Batch size: 8
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001
- Epochs: 4

Modelo 4:

- Batch size: 4 (con acumulación de 4, equivalente a 4x4=16 de batch size)
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001
- Epochs: 3

Modelo 5:

- Primera etapa:
  - Batch size: 8

    - Leaning rate: 1 x 10^{-5}
    - Weight decay: 0.001
    - Epochs: 3
- Segunda etapa:
    - Batch size: 8
    - Leaning rate: 1 x 10^{-5}
    - Weight decay: 0.001
    - Epochs: 6

Modelo 6:

- Primera etapa:
    - Batch size: 8
    - Leaning rate: 1 x 10^{-5}
    - Weight decay: 0.001
    - Epochs: 3
- Segunda etapa:
    - Batch size: 8
    - Leaning rate: 1 x 10^{-5}
    - Weight decay: 0.001
    - Epochs: 6

Modelo 7:

- Batch size: 14
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001
- Epochs: 6

Modelo 8:

- Batch size: 2 (con acumulación de 4, equivalente a 2x4=8 de batch size)
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001

- Epochs: 5

Modelo 9:

- Batch size: 2
- Leaning rate: 1 x 10^{-5}
- Weight decay: 0.001
- Epochs: 8

Modelo Ensemble:

- Como objetivo del modelo se asigna ‘multi:softmax’
- Se realiza una pequeña búsqueda aleatoria de los mejores hiperparámetros con una validación cruzada de 5 y 300 iteraciones. Resultado:
  - subsample: 0.5
  - min_child_weight: 5
  - max_depth: 7
  - 'learning_rate: 0.01
  - colsample_bytree: 0.5

Anexo 2. Resultado de la evaluación de los modelos

Modelo 1 (Run 5):

key: all-accuracy value: 0.6386

key: all-macro f1-score value: 0.5377

key: all-macro precision value: 0.5378

key: all-macro recall value: 0.5438

key: all-micro f1-score value: 0.6386

key: all-micro precision value: 0.6386

key: all-micro recall value: 0.6386

key: de-accuracy value: 0.5967

key: de-macro f1-score value: 0.4671

key: de-macro precision value: 0.4764

key: de-macro recall value: 0.4874

key: de-micro f1-score value: 0.5967

key: de-micro precision value: 0.5967

key: de-micro recall value: 0.5967

key: en-accuracy value: 0.6489

key: en-macro f1-score value: 0.5674

key: en-macro precision value: 0.5696

key: en-macro recall value: 0.567

key: en-micro f1-score value: 0.6489

key: en-micro precision value: 0.6489

key: en-micro recall value: 0.6489

key: es-accuracy value: 0.6403

key: es-macro f1-score value: 0.4188

key: es-macro precision value: 0.4357

key: es-macro recall value: 0.4095

key: es-micro f1-score value: 0.6403

key: es-micro precision value: 0.6403

key: es-micro recall value: 0.6403

key: fr-accuracy value: 0.6567

key: fr-macro f1-score value: 0.536

key: fr-macro precision value: 0.5259

key: fr-macro recall value: 0.5577

key: fr-micro f1-score value: 0.6567

key: fr-micro precision value: 0.6567

key: fr-micro recall value: 0.6567

Modelo 2:

key: all-accuracy value: 0.6994

key: all-macro f1-score value: 0.6127

key: all-macro precision value: 0.6143

key: all-macro recall value: 0.6138

key: all-micro f1-score value: 0.6994

key: all-micro precision value: 0.6994

key: all-micro recall value: 0.6994

key: de-accuracy value: 0.69

key: de-macro f1-score value: 0.5198

key: de-macro precision value: 0.5182

key: de-macro recall value: 0.5238

key: de-micro f1-score value: 0.69

key: de-micro precision value: 0.69

key: de-micro recall value: 0.69

key: en-accuracy value: 0.6963

key: en-macro f1-score value: 0.6315

key: en-macro precision value: 0.6344

key: en-macro recall value: 0.6326

key: en-micro f1-score value: 0.6963

key: en-micro precision value: 0.6963

key: en-micro recall value: 0.6963

key: es-accuracy value: 0.7122

key: es-macro f1-score value: 0.4727

key: es-macro precision value: 0.473

key: es-macro recall value: 0.4727

key: es-micro f1-score value: 0.7122

key: es-micro precision value: 0.7122

key: es-micro recall value: 0.7122

key: fr-accuracy value: 0.71

key: fr-macro f1-score value: 0.5973

key: fr-macro precision value: 0.5979

key: fr-macro recall value: 0.6005

key: fr-micro f1-score value: 0.71

key: fr-micro precision value: 0.71

key: fr-micro recall value: 0.71

Modelo 3:

key: all-accuracy value: 0.5523

key: all-macro f1-score value: 0.3889

key: all-macro precision value: 0.6391

key: all-macro recall value: 0.5625

key: all-micro f1-score value: 0.5523

key: all-micro precision value: 0.5523

key: all-micro recall value: 0.5523

key: de-accuracy value: 0.4

key: de-macro f1-score value: 0.299

key: de-macro precision value: 0.5467

key: de-macro recall value: 0.563

key: de-micro f1-score value: 0.4

key: de-micro precision value: 0.4

key: de-micro recall value: 0.4

key: en-accuracy value: 0.56

key: en-macro f1-score value: 0.4124

key: en-macro precision value: 0.6603

key: en-macro recall value: 0.5762

key: en-micro f1-score value: 0.56

key: en-micro precision value: 0.56

key: en-micro recall value: 0.56

key: es-accuracy value: 0.6331

key: es-macro f1-score value: 0.4138

key: es-macro precision value: 0.6624

key: es-macro recall value: 0.5922

key: es-micro f1-score value: 0.6331

key: es-micro precision value: 0.6331

key: es-micro recall value: 0.6331

key: fr-accuracy value: 0.65

key: fr-macro f1-score value: 0.4001

key: fr-macro precision value: 0.6724

key: fr-macro recall value: 0.5132

key: fr-micro f1-score value: 0.65

key: fr-micro precision value: 0.65

key: fr-micro recall value: 0.65

Modelo 4:

key: all-accuracy value: 0.5495

key: all-macro f1-score value: 0.3888

key: all-macro precision value: 0.6388

key: all-macro recall value: 0.5607

key: all-micro f1-score value: 0.5495

key: all-micro precision value: 0.5495

key: all-micro recall value: 0.5495

key: de-accuracy value: 0.3767

key: de-macro f1-score value: 0.271

key: de-macro precision value: 0.5269

key: de-macro recall value: 0.4889

key: de-micro f1-score value: 0.3767

key: de-micro precision value: 0.3767

key: de-micro recall value: 0.3767

key: en-accuracy value: 0.5585

key: en-macro f1-score value: 0.4113

key: en-macro precision value: 0.6592

key: en-macro recall value: 0.579

key: en-micro f1-score value: 0.5585

key: en-micro precision value: 0.5585

key: en-micro recall value: 0.5585

key: es-accuracy value: 0.6475

key: es-macro f1-score value: 0.4148

key: es-macro precision value: 0.6702

key: es-macro recall value: 0.548

key: es-micro f1-score value: 0.6475

key: es-micro precision value: 0.6475

key: es-micro recall value: 0.6475

key: fr-accuracy value: 0.6567

key: fr-macro f1-score value: 0.4242

key: fr-macro precision value: 0.6896

key: fr-macro recall value: 0.5515

key: fr-micro f1-score value: 0.6567

key: fr-micro precision value: 0.6567

key: fr-micro recall value: 0.6567

Modelo 5 (Run 6):

key: all-accuracy value: 0.6648

key: all-macro f1-score value: 0.6035

key: all-macro precision value: 0.5988

key: all-macro recall value: 0.6091

key: all-micro f1-score value: 0.6648

key: all-micro precision value: 0.6648

key: all-micro recall value: 0.6648

key: de-accuracy value: 0.5867

key: de-macro f1-score value: 0.5411

key: de-macro precision value: 0.5244

key: de-macro recall value: 0.6306

key: de-micro f1-score value: 0.5867

key: de-micro precision value: 0.5867

key: de-micro recall value: 0.5867

key: en-accuracy value: 0.6933

key: en-macro f1-score value: 0.6459

key: en-macro precision value: 0.6558

key: en-macro recall value: 0.6378

key: en-micro f1-score value: 0.6933

key: en-micro precision value: 0.6933

key: en-micro recall value: 0.6933

key: es-accuracy value: 0.6906

key: es-macro f1-score value: 0.5026

key: es-macro precision value: 0.5099

key: es-macro recall value: 0.4987

key: es-micro f1-score value: 0.6906

key: es-micro precision value: 0.6906

key: es-micro recall value: 0.6906

key: fr-accuracy value: 0.6667

key: fr-macro f1-score value: 0.5754

key: fr-macro precision value: 0.5748

key: fr-macro recall value: 0.6007

key: fr-micro f1-score value: 0.6667

key: fr-micro precision value: 0.6667

key: fr-micro recall value: 0.6667

Modelo 6:

key: all-accuracy value: 0.6393

key: all-macro f1-score value: 0.5545

key: all-macro precision value: 0.5545

key: all-macro recall value: 0.5547

key: all-micro f1-score value: 0.6393

key: all-micro precision value: 0.6393

key: all-micro recall value: 0.6393

key: de-accuracy value: 0.5933

key: de-macro f1-score value: 0.505

key: de-macro precision value: 0.4987

key: de-macro recall value: 0.517

key: de-micro f1-score value: 0.5933

key: de-micro precision value: 0.5933

key: de-micro recall value: 0.5933

key: en-accuracy value: 0.643

key: en-macro f1-score value: 0.5781

key: en-macro precision value: 0.577

key: en-macro recall value: 0.5793

key: en-micro f1-score value: 0.643

key: en-micro precision value: 0.643

key: en-micro recall value: 0.643

key: es-accuracy value: 0.7122

key: es-macro f1-score value: 0.4692

key: es-macro precision value: 0.4669

key: es-macro recall value: 0.4727

key: es-micro f1-score value: 0.7122

key: es-micro precision value: 0.7122

key: es-micro recall value: 0.7122

key: fr-accuracy value: 0.6433

key: fr-macro f1-score value: 0.489

key: fr-macro precision value: 0.4899

key: fr-macro recall value: 0.4927

key: fr-micro f1-score value: 0.6433

key: fr-micro precision value: 0.6433

key: fr-micro recall value: 0.6433

Modelo 7 (Run 2):

key: all-accuracy value: 0.6765

key: all-macro f1-score value: 0.5969

key: all-macro precision value: 0.582

key: all-macro recall value: 0.619

key: all-micro f1-score value: 0.6765

key: all-micro precision value: 0.6765

key: all-micro recall value: 0.6765

key: bg-accuracy value: 0.0

key: bg-macro f1-score value: 0.0

key: bg-macro precision value: 0.3333

key: bg-macro recall value: 0.0

key: bg-micro f1-score value: 0.0

key: bg-micro precision value: 0.0

key: bg-micro recall value: 0.0

key: ca-accuracy value: 0.9167

key: ca-macro f1-score value: 0.8737

key: ca-macro precision value: 0.95

key: ca-macro recall value: 0.8333

key: ca-micro f1-score value: 0.9167

key: ca-micro precision value: 0.9167

key: ca-micro recall value: 0.9167

key: cs-accuracy value: 0.5625

key: cs-macro f1-score value: 0.3556

key: cs-macro precision value: 0.4167

key: cs-macro recall value: 0.3545

key: cs-micro f1-score value: 0.5625

key: cs-micro precision value: 0.5625

key: cs-micro recall value: 0.5625

key: da-accuracy value: 1.0

key: da-macro f1-score value: 1.0

key: da-macro precision value: 1.0

key: da-macro recall value: 1.0

key: da-micro f1-score value: 1.0

key: da-micro precision value: 1.0

key: da-micro recall value: 1.0

key: de-accuracy value: 0.6689

key: de-macro f1-score value: 0.5086

key: de-macro precision value: 0.5069

key: de-macro recall value: 0.5121

key: de-micro f1-score value: 0.6689

key: de-micro precision value: 0.6689

key: de-micro recall value: 0.6689

key: el-accuracy value: 0.875

key: el-macro f1-score value: 0.4667

key: el-macro precision value: 0.9375

key: el-macro recall value: 0.5

key: el-micro f1-score value: 0.875

key: el-micro precision value: 0.875

key: el-micro recall value: 0.875

key: en-accuracy value: 0.6703

key: en-macro f1-score value: 0.6128

key: en-macro precision value: 0.5906

key: en-macro recall value: 0.6499

key: en-micro f1-score value: 0.6703

key: en-micro precision value: 0.6703

key: en-micro recall value: 0.6703

key: eo-accuracy value: 0.7778

key: eo-macro f1-score value: 0.7231

key: eo-macro precision value: 0.7083

key: eo-macro recall value: 0.7679

key: eo-micro f1-score value: 0.7778

key: eo-micro precision value: 0.7778

key: eo-micro recall value: 0.7778

key: es-accuracy value: 0.7216

key: es-macro f1-score value: 0.6119

key: es-macro precision value: 0.594

key: es-macro recall value: 0.6495

key: es-micro f1-score value: 0.7216

key: es-micro precision value: 0.7216

key: es-micro recall value: 0.7216

key: fi-accuracy value: 0.6667

key: fi-macro f1-score value: 0.7222

key: fi-macro precision value: 0.8333

key: fi-macro recall value: 0.7778

key: fi-micro f1-score value: 0.6667

key: fi-micro precision value: 0.6667

key: fi-micro recall value: 0.6667

key: fr-accuracy value: 0.6598

key: fr-macro f1-score value: 0.5394

key: fr-macro precision value: 0.5344

key: fr-macro recall value: 0.5458

key: fr-micro f1-score value: 0.6598

key: fr-micro precision value: 0.6598

key: fr-micro recall value: 0.6598

key: hu-accuracy value: 0.7273

key: hu-macro f1-score value: 0.7059

key: hu-macro precision value: 0.7083

key: hu-macro recall value: 0.7089

key: hu-micro f1-score value: 0.7273

key: hu-micro precision value: 0.7273

key: hu-micro recall value: 0.7273

key: it-accuracy value: 0.7018

key: it-macro f1-score value: 0.51

key: it-macro precision value: 0.5157

key: it-macro recall value: 0.5136

key: it-micro f1-score value: 0.7018

key: it-micro precision value: 0.7018

key: it-micro recall value: 0.7018

key: lt-accuracy value: 0.6667

key: lt-macro f1-score value: 0.4722

key: lt-macro precision value: 0.5833

key: lt-macro recall value: 0.4167

key: lt-micro f1-score value: 0.6667

key: lt-micro precision value: 0.6667

key: lt-micro recall value: 0.6667

key: nl-accuracy value: 0.6792

key: nl-macro f1-score value: 0.5648

key: nl-macro precision value: 0.5692

key: nl-macro recall value: 0.5622

key: nl-micro f1-score value: 0.6792

key: nl-micro precision value: 0.6792

key: nl-micro recall value: 0.6792

key: pl-accuracy value: 0.3846

key: pl-macro f1-score value: 0.2751

key: pl-macro precision value: 0.2952

key: pl-macro recall value: 0.2917

key: pl-micro f1-score value: 0.3846

key: pl-micro precision value: 0.3846

key: pl-micro recall value: 0.3846

key: pt-accuracy value: 0.8462

key: pt-macro f1-score value: 0.7833

key: pt-macro precision value: 0.9091

key: pt-macro recall value: 0.75

key: pt-micro f1-score value: 0.8462

key: pt-micro precision value: 0.8462

key: pt-micro recall value: 0.8462

key: ro-accuracy value: 0.8462

key: ro-macro f1-score value: 0.3188

key: ro-macro precision value: 0.6667

key: ro-macro recall value: 0.3056

key: ro-micro f1-score value: 0.8462

key: ro-micro precision value: 0.8462

key: ro-micro recall value: 0.8462

key: sk-accuracy value: 0.5

key: sk-macro f1-score value: 0.25

key: sk-macro precision value: 0.25

key: sk-macro recall value: 0.25

key: sk-micro f1-score value: 0.5

key: sk-micro precision value: 0.5

key: sk-micro recall value: 0.5

key: sl-accuracy value: 1.0

key: sl-macro f1-score value: 1.0

key: sl-macro precision value: 1.0

key: sl-macro recall value: 1.0

key: sl-micro f1-score value: 1.0

key: sl-micro precision value: 1.0

key: sl-micro recall value: 1.0

key: sv-accuracy value: 0.6364

key: sv-macro f1-score value: 0.4167

key: sv-macro precision value: 0.4524

key: sv-macro recall value: 0.3889

key: sv-micro f1-score value: 0.6364

key: sv-micro precision value: 0.6364

key: sv-micro recall value: 0.6364

Modelo 8 (Run 3):

key: all-accuracy value: 0.7108

key: all-macro f1-score value: 0.6158

key: all-macro precision value: 0.6207

key: all-macro recall value: 0.6123

key: all-micro f1-score value: 0.7108

key: all-micro precision value: 0.7108

key: all-micro recall value: 0.7108

key: bg-accuracy value: 0.75

key: bg-macro f1-score value: 0.7333

key: bg-macro precision value: 0.8333

key: bg-macro recall value: 0.75

key: bg-micro f1-score value: 0.75

key: bg-micro precision value: 0.75

key: bg-micro recall value: 0.75

key: ca-accuracy value: 0.8182

key: ca-macro f1-score value: 0.3158

key: ca-macro precision value: 0.3333

key: ca-macro recall value: 0.3

key: ca-micro f1-score value: 0.8182

key: ca-micro precision value: 0.8182

key: ca-micro recall value: 0.8182

key: cs-accuracy value: 0.9167

key: cs-macro f1-score value: 0.641

key: cs-macro precision value: 0.6667

key: cs-macro recall value: 0.619

key: cs-micro f1-score value: 0.9167

key: cs-micro precision value: 0.9167

key: cs-micro recall value: 0.9167

key: da-accuracy value: 0.5

key: da-macro f1-score value: 0.3333

key: da-macro precision value: 0.5

key: da-macro recall value: 0.25

key: da-micro f1-score value: 0.5

key: da-micro precision value: 0.5

key: da-micro recall value: 0.5

key: de-accuracy value: 0.7429

key: de-macro f1-score value: 0.6738

key: de-macro precision value: 0.6712

key: de-macro recall value: 0.6768

key: de-micro f1-score value: 0.7429

key: de-micro precision value: 0.7429

key: de-micro recall value: 0.7429

key: el-accuracy value: 0.8

key: el-macro f1-score value: 0.7847

key: el-macro precision value: 0.7778

key: el-macro recall value: 0.8

key: el-micro f1-score value: 0.8

key: el-micro precision value: 0.8

key: el-micro recall value: 0.8

key: en-accuracy value: 0.6671

key: en-macro f1-score value: 0.58

key: en-macro precision value: 0.5836

key: en-macro recall value: 0.5776

key: en-micro f1-score value: 0.6671

key: en-micro precision value: 0.6671

key: en-micro recall value: 0.6671

key: eo-accuracy value: 0.6364

key: eo-macro f1-score value: 0.2593

key: eo-macro precision value: 0.5667

key: eo-macro recall value: 0.2917

key: eo-micro f1-score value: 0.6364

key: eo-micro precision value: 0.6364

key: eo-micro recall value: 0.6364

key: es-accuracy value: 0.7755

key: es-macro f1-score value: 0.6351

key: es-macro precision value: 0.6348

key: es-macro recall value: 0.6355

key: es-micro f1-score value: 0.7755

key: es-micro precision value: 0.7755

key: es-micro recall value: 0.7755

key: fi-accuracy value: 0.6667

key: fi-macro f1-score value: 0.6494

key: fi-macro precision value: 0.6667

key: fi-macro recall value: 0.65

key: fi-micro f1-score value: 0.6667

key: fi-micro precision value: 0.6667

key: fi-micro recall value: 0.6667

key: fr-accuracy value: 0.7396

key: fr-macro f1-score value: 0.5979

key: fr-macro precision value: 0.6515

key: fr-macro recall value: 0.5724

key: fr-micro f1-score value: 0.7396

key: fr-micro precision value: 0.7396

key: fr-micro recall value: 0.7396

key: ga-accuracy value: 0.6667

key: ga-macro f1-score value: 0.6667

key: ga-macro precision value: 0.75

key: ga-macro recall value: 0.75

key: ga-micro f1-score value: 0.6667

key: ga-micro precision value: 0.6667

key: ga-micro recall value: 0.6667

key: hr-accuracy value: 1.0

key: hr-macro f1-score value: 1.0

key: hr-macro precision value: 1.0

key: hr-macro recall value: 1.0

key: hr-micro f1-score value: 1.0

key: hr-micro precision value: 1.0

key: hr-micro recall value: 1.0

key: hu-accuracy value: 0.7907

key: hu-macro f1-score value: 0.7269

key: hu-macro precision value: 0.7129

key: hu-macro recall value: 0.7508

key: hu-micro f1-score value: 0.7907

key: hu-micro precision value: 0.7907

key: hu-micro recall value: 0.7907

key: it-accuracy value: 0.6833

key: it-macro f1-score value: 0.5373

key: it-macro precision value: 0.5417

key: it-macro recall value: 0.5483

key: it-micro f1-score value: 0.6833

key: it-micro precision value: 0.6833

key: it-micro recall value: 0.6833

key: lt-accuracy value: 0.8333

key: lt-macro f1-score value: 0.7778

key: lt-macro precision value: 0.9

key: lt-macro recall value: 0.75

key: lt-micro f1-score value: 0.8333

key: lt-micro precision value: 0.8333

key: lt-micro recall value: 0.8333

key: nl-accuracy value: 0.7903

key: nl-macro f1-score value: 0.5214

key: nl-macro precision value: 0.5644

key: nl-macro recall value: 0.5161

key: nl-micro f1-score value: 0.7903

key: nl-micro precision value: 0.7903

key: nl-micro recall value: 0.7903

key: pl-accuracy value: 0.7143

key: pl-macro f1-score value: 0.5079

key: pl-macro precision value: 0.5833

key: pl-macro recall value: 0.5

key: pl-micro f1-score value: 0.7143

key: pl-micro precision value: 0.7143

key: pl-micro recall value: 0.7143

key: pt-accuracy value: 1.0

key: pt-macro f1-score value: 1.0

key: pt-macro precision value: 1.0

key: pt-macro recall value: 1.0

key: pt-micro f1-score value: 1.0

key: pt-micro precision value: 1.0

key: pt-micro recall value: 1.0

key: ro-accuracy value: 0.7647

key: ro-macro f1-score value: 0.5952

key: ro-macro precision value: 0.65

key: ro-macro recall value: 0.5865

key: ro-micro f1-score value: 0.7647

key: ro-micro precision value: 0.7647

key: ro-micro recall value: 0.7647

key: sk-accuracy value: 0.7273

key: sk-macro f1-score value: 0.6857

key: sk-macro precision value: 0.6786

key: sk-macro recall value: 0.7083

key: sk-micro f1-score value: 0.7273

key: sk-micro precision value: 0.7273

key: sk-micro recall value: 0.7273

key: sl-accuracy value: 1.0

key: sl-macro f1-score value: 1.0

key: sl-macro precision value: 1.0

key: sl-macro recall value: 1.0

key: sl-micro f1-score value: 1.0

key: sl-micro precision value: 1.0

key: sl-micro recall value: 1.0

key: sv-accuracy value: 0.75

key: sv-macro f1-score value: 0.4286

key: sv-macro precision value: 0.4286

key: sv-macro recall value: 0.4286

key: sv-micro f1-score value: 0.75

key: sv-micro precision value: 0.75

key: sv-micro recall value: 0.75

Modelo 9 (Run 4):

key: all-accuracy value: 0.6575

key: all-macro f1-score value: 0.5779

key: all-macro precision value: 0.5878

key: all-macro recall value: 0.5868

key: all-micro f1-score value: 0.6575

key: all-micro precision value: 0.6575

key: all-micro recall value: 0.6575

key: de-accuracy value: 0.5862

key: de-macro f1-score value: 0.3857

key: de-macro precision value: 0.3849

key: de-macro recall value: 0.3876

key: de-micro f1-score value: 0.5862

key: de-micro precision value: 0.5862

key: de-micro recall value: 0.5862

key: en-accuracy value: 0.6564

key: en-macro f1-score value: 0.581

key: en-macro precision value: 0.5901

key: en-macro recall value: 0.5916

key: en-micro f1-score value: 0.6564

key: en-micro precision value: 0.6564

key: en-micro recall value: 0.6564

key: es-accuracy value: 0.8462

key: es-macro f1-score value: 0.5744

key: es-macro precision value: 0.5667

key: es-macro recall value: 0.5825

key: es-micro f1-score value: 0.8462

key: es-micro precision value: 0.8462

key: es-micro recall value: 0.8462

key: fr-accuracy value: 0.6721

key: fr-macro f1-score value: 0.4738

key: fr-macro precision value: 0.6288

key: fr-macro recall value: 0.4663

key: fr-micro f1-score value: 0.6721

key: fr-micro precision value: 0.6721

key: fr-micro recall value: 0.6721

Modelo XGboost (Run 1):

key: all-accuracy value: 0.735

key: all-macro f1-score value: 0.6691

key: all-macro precision value: 0.6835

key: all-macro recall value: 0.6581

key: all-micro f1-score value: 0.735

key: all-micro precision value: 0.735

key: all-micro recall value: 0.735

key: de-accuracy value: 0.6984

key: de-macro f1-score value: 0.676

key: de-macro precision value: 0.6404

key: de-macro recall value: 0.7731

key: de-micro f1-score value: 0.6984

key: de-micro precision value: 0.6984

key: de-micro recall value: 0.6984

key: en-accuracy value: 0.7581

key: en-macro f1-score value: 0.68

key: en-macro precision value: 0.7126

key: en-macro recall value: 0.6621

key: en-micro f1-score value: 0.7581

key: en-micro precision value: 0.7581

key: en-micro recall value: 0.7581

key: es-accuracy value: 0.7419

key: es-macro f1-score value: 0.4957

key: es-macro precision value: 0.8121

key: es-macro recall value: 0.514

key: es-micro f1-score value: 0.7419

key: es-micro precision value: 0.7419

key: es-micro recall value: 0.7419

key: fr-accuracy value: 0.7231

key: fr-macro f1-score value: 0.6012

key: fr-macro precision value: 0.6233

key: fr-macro recall value: 0.5862

key: fr-micro f1-score value: 0.7231

key: fr-micro precision value: 0.7231

key: fr-micro recall value: 0.7231