DETECCIÓN AUTOMÁTICA DE IDIOMAS EN TEXTOS CORTOS DE REPOSITORIOS INSTITUCIONALES:

¿vale la pena realizar un ajuste fino sobre un modelo de lenguaje?

Carlos Javier Nusch*
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
carlosnusch@sedici.unlp.edu.ar
https://orcid.org/0000-0003-1715-4228

Leandro Antonelli
Universidad Nacional de La Plata;
LIFIA, Argentina
lanto@lifia.info.unlp.edu.ar
https://orcid.org/0000-0003-1388-0337

Leticia Cecilia Cagnina
Universidad Nacional de San Luis;
LIDIC, Argentina
lcagnina@unsl.edu.ar
https://orcid.org/0000-0001-7825-2927

Marcelo Luis Errecalde
Universidad Nacional de San Luis;
LIDIC, Argentina
merreca@unsl.edu.ar
https://orcid.org/0000-0001-5605-8963

Ariel Lira
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
alira@sedici.unlp.edu
https://orcid.org/0000-0003-3647-3101

Marisa Raquel De Giusti
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
marisadegiusti@gmail.com
https://orcid.org/ 0000-0003-2422-6322

Gonzalo Villarreal
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
gonzalo@prebi.unlp.edu.ar
https://orcid.org/0000-0002-3602-8211

Santiago Tettamanti
Universidad Nacional de La Plata;
PREBI-SEDICI, Argentina
stettamanti@indec.gob.ar
https://orcid.org/0000-0003-3339-7940

DOI: *10.22477/xiv.biredial.412*
EJE TEMÁICO: *Infraestructura tecnológica*

RESUMEN

Presentación del problema: Este artículo busca continuar y optimizar las tareas de detección automática de idioma llevadas a cabo previamente en el repositorio institucional SEDICI. Se procura facilitar la catalogación de materiales ante el enorme volumen de recursos almacenados actualmente. **Materiales y metodología:** A partir de un dataset exportado del repositorio de unos 126.081 ítems se planificó una tarea de detección automática

* Las contribuciones de los autores fueron declaradas siguiendo la taxonomía TaDiRAH (Taxonomy of Digital Research Activities in the Humanities). Carlos Javier Nusch fue responsable del desarrollo del código, el procesamiento del lenguaje natural, el análisis de datos y la redacción del artículo (programación, modelado, codificación de texto, análisis, limpieza, evaluación y enriquecimiento). Leticia Cagnina, Ariel Lira, Gonzalo Villarreal, Leandro Antonelli, Marcelo Errecalde y Marisa De Giusti participaron en tareas de revisión y corrección del contenido. Santiago Tettamanti estuvo a cargo de la recolección y curaduría del dataset original de 2022.

de idiomas utilizando diferentes bibliotecas existentes con el enfoque zero-shot (LangDetect, Polyglot y Langid). Previamente se llevaron a cabo varias tareas de limpieza de texto y preprocesamiento que buscaron mejorar el desempeño de las bibliotecas respecto de tareas anteriores. Luego se compararon los resultados obtenidos con los datos de los idiomas registrados por el personal de catalogación del repositorio y se corroboró la exactitud de uno y otro grupo. Para tratar de mejorar aún más la detección de idiomas se realizó un ajuste fino y analizó el desempeño de la biblioteca Fasttext y varios modelos (mBERT, SBERT y XLM-RoBERTa). **Resultados:** En general, todas las bibliotecas de detección de idiomas mostraron un alto nivel de precisión en la detección de idiomas, alrededor de un 98%. En el caso de los modelos de lenguaje también se obtuvieron muy buenos resultados, con valores de alrededor de 100% de f1 score. Las diferentes tareas llevadas a cabo también permitieron identificar y tipificar algunos errores recurrentes en los que suelen incurrir los catalogadores humanos así como realizar una corrección en lote de los metadatos erróneos.

Palabras-clave: Repositorios Institucionales, tareas de curaduría de datos, detección automática de idiomas, mBERT, SBERT, XLM-RoBERTa, enfoque zero-shot, ajuste fino de modelos.

ABSTRACT

Problem Statement: This article aims to continue and optimize the automatic language detection tasks previously carried out in the SEDICI institutional repository. The goal is to facilitate the cataloging of materials given the enormous volume of resources currently stored. **Materials and Methodology:** Starting with a dataset exported from the repository, consisting of about 126,081 items, an automatic language detection task was planned using different existing libraries with a zero-shot approach (LangDetect, Polyglot, and Langid). Several text cleaning and pre-processing tasks were carried out beforehand to improve the libraries' performance compared to previous tasks. Then, the results obtained were compared with the language data registered by the repository's cataloging personnel, and the accuracy of both groups was corroborated. To try to further improve language detection, a fine-tuning process was performed, and the performance of the Fasttext library and several models (mBERT, SBERT, and XLM-RoBERTa) was analyzed. **Results:** In general, all language detection libraries showed a ** high level of accuracy** in language detection, at around 98%. In the case of the language models, very good results were also obtained, with f1 scores of around 100%. The different tasks carried out also made it possible to identify and categorize some recurring errors typically made by human catalogers, as well as to perform a batch correction of the erroneous metadata.)

Keywords: Institutional Repositories, Data Curation Tasks, Automatic Language Detection, mBERT, SBERT, XLM--RoBERTa , zero-shot approach, models fine-tuning.

INTRODUCCIÓN

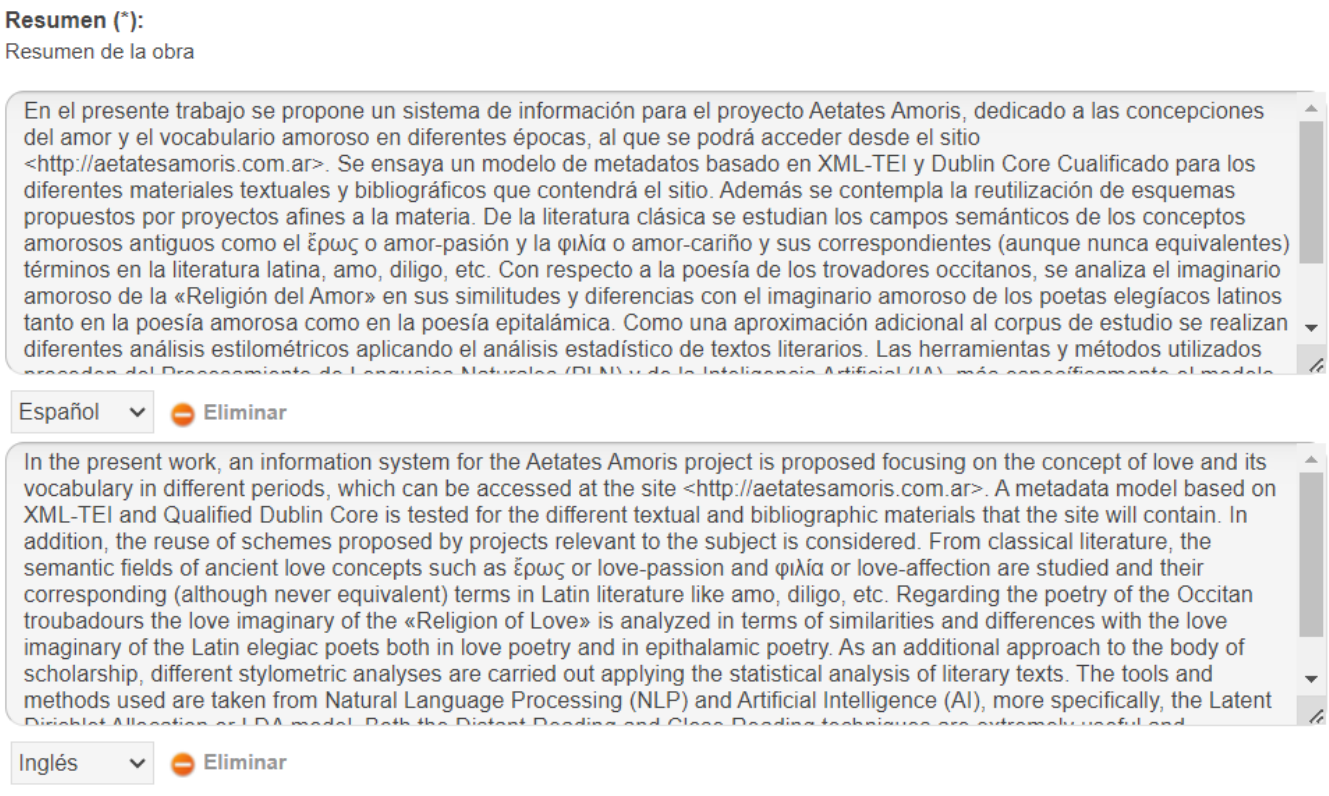
El presente trabajo se enmarca dentro de lo que se conoce como Descubrimiento de Conocimiento en Bases de Datos (KDD, del inglés Knowledge Discovery in Databases) (Fayyad et al., 1996); más comúnmente asociado con la Minería de Datos o extracción de conocimiento e información útiles desde datos crudos. En el caso de la extracción de nueva información y patrones desde de datos de texto se suele denominar Descubrimiento de Conocimiento en Texto (KDT) (Feldman & Dagan, 1995). El ámbito de aplicación de las tareas que se describirán a continuación es el de los repositorios institucionales y de acceso abierto.

El repositorio central de la Universidad Nacional de La Plata, Servicio de Difusión de la Creación Intelectual (SEDICI), cuenta actualmente con 168326 recursos, una cifra enorme que

evidencia un crecimiento exponencial[1] producto de años de arduo trabajo. La asignación del metadato idioma, tanto para el texto completo o los campos destinados al resumen de un ítem es una de las tantas tareas de catalogación llevadas a cabo por el personal que administra el repositorio.

Dada la cantidad de campos que se deben revisar y ajustar en atención a las buenas prácticas, normas y directrices del repositorio (título, autores, resumen, palabras clave, etc.), y que dichos campos deben revisarse en cada uno de los ítems que se procesan a diario existe una alta probabilidad de que se cometan diferentes tipos de errores. El riesgo de cometer errores se acrecienta cada año con el volumen de ítems que ingestan inclusive en tareas automáticas de importación (De Giusti et al., 2016). El campo de idioma para los resúmenes de ítems del repositorio, uno de los tantos que tiene el flujo de trabajo de catalogación e ingesta de material del software DSpace se encuentra generalmente situado debajo del campo de texto del resumen. Por su tamaño, y porque el administrador generalmente suele estar atento a la corrección de varios campos diferentes entre otras tareas simultáneas que se realizan, suele ser un campo propicio a errores. El administrador puede cliquear mal al escoger el idioma con el mouse o bien puede pasar por alto ese campo y el idioma registrado en consecuencia es el valor asignado por defecto, en nuestro caso, el español (Fig. 1).

Figura 1. Vista de los campos de resumen para un catalogador en DSpace. Elaboración propia.



Fuente: elaboración propia.

[1] Datos accesibles desde: http://sedici.unlp.edu.ar/pages/estadisticasContenidoRepositorio

Con la finalidad de explorar el grado de corrección en la catalogación del atributo idioma se exportó un dataset en formato csv el 7 de abril de 2022. El conjunto de datos incluía información de 126.081 ítems, todos los presentes en el repositorio hasta esa fecha. El objetivo original consistía en llevar a cabo una tarea de curaduría automática aprovechando las diferentes herramientas de detección de idiomas disponibles en la actualidad.

ANTECEDENTES Y RESULTADOS ANTERIORES

La detección automática de lenguajes tiene sus orígenes en investigaciones pioneras realizadas en la década de 1990 con enfoques estadísticos aplicados a la clasificación de textos. Cavnar y Trenkle (1994) propusieron un método basado en perfiles de n-gramas de caracteres y comparación por distancia de rankings. También se desarrollaron otras propuestas orientadas en el campo de la síntesis de voz multilingüe que requerían análisis lingüístico previo y empleaban reglas lingüísticas y análisis morfosintáctico (Sproat, 1996). A partir de los años 2000, con el avance del aprendizaje automático y la creciente disponibilidad de corpus multilingües, surgieron herramientas más robustas y precisas como Langid.py (Lui & Baldwin, 2012), basado en un modelo de Naive Bayes o el auge de los clasificadores rápidos basados en embeddings, como FastText (Joulin *et al.*, 2016b). Unos años antes, McNamee (2005) había diseñado un sistema de detección de idiomas usando vectores de frecuencia de *n*-gramas y diversas métricas de similitud. Su análisis destacó que la distancia de Manhattan ofrecía el mejor rendimiento frente a otras alternativas, y propuso este ejercicio como actividad formativa en cursos de inteligencia artificial y procesamiento de texto. Herramientas como LangDetect y CLD3, derivadas de desarrollos internos de Google, popularizaron el uso de histogramas de frecuencias y redes neuronales ligeras para detección de idioma zero-shot (Ooms & Google Inc, 2023; Shuyo, 2010).

En el último lustro se han realizado avances significativos, especialmente gracias a la integración de modelos de lenguaje preentrenados y técnicas de aprendizaje profundo adaptadas a contextos multilingües. Uno de los trabajos más relevantes en este período es el de Caswell *et al.* (2020), quienes desarrollaron un sistema basado en arquitecturas *Transformer* capaz de identificar más de 212 lenguas en datos web. Desde una perspectiva comparativa, Jauhiainen *et al.* (2021) analizaron el desempeño de enfoques estadísticos adaptativos (como Naive Bayes y el sistema HeLI 2.0) frente a modelos basados en *transformers* (mBERT, XLM-R) en escenarios de alta complejidad, específicamente para lenguas dravídicas, habladas principalmente en el sur de la India. Adebara *et al.* (2022, 2023) propusieron y evaluaron modelos multilingües especializados para lenguas africanas, como AfroLID y SERENGETI. Estos modelos demostraron mejoras significativas frente a variantes genéricas como mBERT, especialmente en idiomas subrepresentados.

En cuanto al procesamiento y clasificación de idiomas en textos cortos Balazevic *et al.* (2016) abordaron la identificación automática de idioma en textos extremadamente breves,

como los mensajes de Twitter, proponiendo un modelo probabilístico de *n*-gramas de caracteres mejorado con información contextual del usuario. Más recientemente, Marivate y Sefara (2020) propusieron estrategias de aumento de datos globales que mejoran la clasificación en textos breves mediante síntesis y perturbación de datos, fortaleciendo la capacidad de generalización de los clasificadores en modelos de aprendizaje profundo. Desde una perspectiva orientada a la curación de datos masivos, Bañón *et al.* (2024) desarrollaron FastSpell, un sistema híbrido que combina un clasificador rápido con verificación ortográfica mediante diccionarios Hunspell, orientado específicamente a reducir errores sistemáticos en lenguas próximas y variantes dialectales como el gallego, catalán o el serbio-croata.

A su vez, el trabajo de Sindane y Marivate (2024) representa una contribución clave en el campo de las lenguas subrepresentadas. Evaluaron desde enfoques clásicos de aprendizaje automático basados en *n*-gramas hasta modelos multilingües de última generación como mBERT, XLM-R, AfriBERTa y SERENGETI, focalizándose en once lenguas sudafricanas. Si bien los modelos basados en arquitecturas Transformer obtuvieron los mejores resultados globales (con SERENGETI superando el 98 % de precisión), los clasificadores ligeros como Naive Bayes y el modelo compacto *za_BERT_lid* entrenado por los autores demostraron ser altamente competitivos, con costos computacionales mucho menores.

En cuanto al ámbito académico y científico, Céspedes *et al.* (2025) realizaron un análisis crítico de la calidad de los metadatos lingüísticos en OpenAlex[2] entre los que se encontraba la cobertura de diferentes idiomas para sus metadatos. Los resultados revelaron una cobertura más diversa que en bases comerciales como Web of Science[3], pero también una alta tasa de errores y ausencias en la identificación de idioma, lo que pone en evidencia la necesidad de soluciones más robustas y adaptativas. En respuesta directa a esta problemática, Holmberg *et al.* (2025) evaluaron el rendimiento de múltiples herramientas de detección lingüística —incluidas LangID, FastText, LangDetect, CLD3 y FastSpell— sobre miles de registros multilingües de OpenAlex, con especial atención a las condiciones reales del repositorio (longitudes variables, metadatos incompletos). Los autores propusieron un enfoque híbrido multi-etapa que combina clasificadores probabilísticos rápidos con verificación contextual, y concluyó que modelos como FastSpell operando sobre campos breves (por ejemplo, títulos de artículos) ofrecían el mejor equilibrio entre precisión, recuperación y velocidad de procesamiento para tareas de detección de idioma a gran escala.

En este escenario de avances metodológicos y crecientes exigencias de calidad en la curaduría de datos multilingües, el presente trabajo se inscribe como una contribución aplicada centrada en un corpus institucional real y en la evaluación práctica del ajuste fino de mode-

[2] OpenAlex es una base de datos abierta y de acceso libre que indexa publicaciones académicas, autores, instituciones, revistas y conceptos. Disponible en: https://openalex.org/
[3] Web of Science (WoS) es una plataforma bibliográfica comercial mantenida por Clarivate Analytics, ampliamente utilizada para la indexación, análisis y evaluación de la producción científica. Disponible en: https://www.webofscience.com/wos/

los de lenguaje sobre textos breves, como los resúmenes depositados en SEDICI. En el ámbito de los repositorios institucionales y enfocado en la detección de idiomas en textos cortos no se han detectado trabajos específicos salvo el realizado anteriormente (Nusch et al., 2025) en el que se utilizaron las bibliotecas Langid[4], CLD3[5], Textcat[6], Polyglot[7], FastText[8] y Langdetect[9]. En todos los casos, las bibliotecas mostraron un desempeño cercano al 95% comparado con los idiomas identificados por catalogadores humanos, excepto FastText, que obtuvo peores resultados.

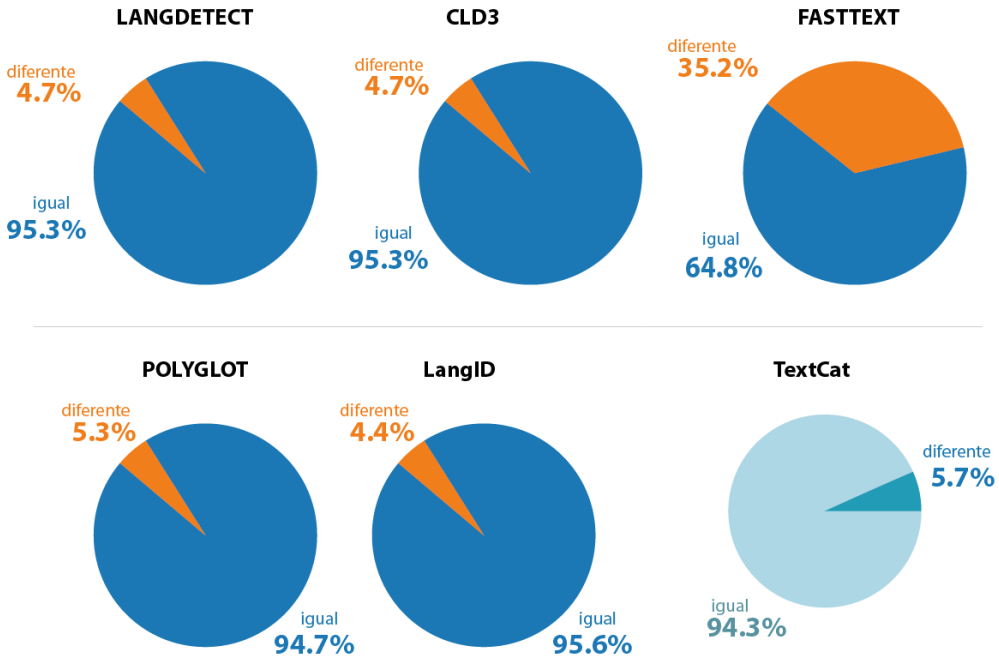
Tabla 1 - Porcentaje de coincidencias en la detección de idiomas y desempeño de diferentes bibliotecas (Nusch et al., 2025).

Biblioteca	Igual al catalogador humano	Diferente al catalogador humano	Tiempo de ejecución
langdetect	95.3	4.7	25 mins 9.53 secs
CLD3	95.3	4.7	3 mins 56.60 secs
fastText	64.8	35.2	**2 mins 5.02 secs**
Polyglot	94.7	5.3	2 mins 37.24 secs
langid	**95.6**	4.4	13 mins 42.24 secs
TextCat	94.3	5.7	2 hours, 2 mins 39 secs[10]

Fuente: elaboración propia.

[4] Disponible en: https://github.com/saffsd/langid.py
[5] Disponible en: https://github.com/ropensci/cld3
[6] Disponible en: https://cran.r-project.org/web/packages/textcat
[7] Disponible en: https://github.com/saffsd/polyglot
[8] Disponible en: https://fasttext.cc/
[9] Disponible en: https://pypi.org/project/langdetect/
[10] La discrepancia entre los tiempos de las otras bibliotecas y TextCat se debió en aquel caso a que esta última fue ejecutada en una computadora local en R Studio mientras que las anteriores se corrieron en Google Colab con el lenguaje Python.

Figura 2. Gráficos de torta con el porcentaje de coincidencia de los idiomas detectados por cada biblioteca comparado con los idiomas catalogados por humanos (Nusch et al., 2025). Elaboración propia.



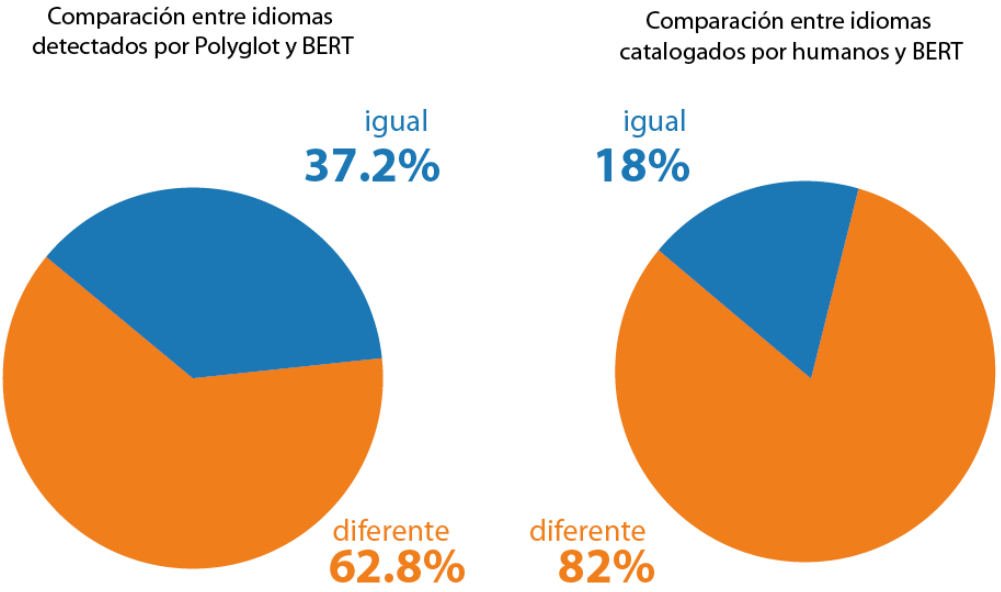
Fuente: elaboración propia.

También se evaluó un modelo mBERT, el cual presentó dificultades para clasificar correctamente ciertos idiomas. En particular, el español fue confundido con inglés y francés, mientras que el modelo mostró un bajo desempeño en la identificación de italiano, así como resultados limitados en las clases de francés y alemán.

Se estableció entonces un proceso de mejora en el desempeño del modelo mBERT por medio de la estratificación de los datos de entrenamiento y varias tareas de aumento de datos para las clases minoritarias utilizando el modelo MarianMT[11] optimizado para la traducción de textos. La tarea de aumento de datos consistió en la traducción de resúmenes de las clases mayoritarias de idioma a las clases minoritarias con el fin de tener un dataset más equilibrado. Tras el aumento de datos, se hizo un ajuste fino sobre el modelo mBERT y se generaron métricas de evaluación satisfactorias con precisiones cercanas a 1 (100%) para varios idiomas y una exactitud total del modelo en el conjunto de validación fue de 0.999 (99.9%), lo que indica un rendimiento notable después de las mejoras implementadas. Luego se comparó el modelo para detectar idiomas en la parte del dataset que había presentado alguna diferencia entre las bibliotecas y los catalogadores humanos utilizando como referencica lo detectado por la biblioteca Polyglot.

[11] Disponible en: https://huggingface.co/docs/transformers/model_doc/marian

Figura 3. Gráficos de torta con el porcentaje de coincidencia de los idiomas por mBERT comparado con los idiomas detectados por Polyglot y con los idiomas catalogados por humanos (Nusch et al., 2025).



Fuente: elaboración propia.

En aquel momento, debido al gran tamaño de la colección, no se corroboró la exactitud de la catalogación de idiomas en los casos en los que los humanos y las bibliotecas no coincidían (alrededor de unos 3000 ítems, algunos con más de un resumen), tarea que se llevó a cabo finalmente para el presente artículo.

NUEVAS TAREAS LLEVADAS A CABO PARA LA MEJORA DEL DESEMPEÑO DE MODELOS Y BIBLIOTECAS

BIBLIOTECAS PARA LA DETECCIÓN AUTOMÁTICA DE IDIOMAS: ENFOQUE *ZERO-SHOT*[12]

Al igual que en el artículo anterior, se utilizó el lenguaje Python y el mismo dataset; se analizaron los campos de textos de los resúmenes de los diferentes ítems y las etiquetas de idioma aplicadas sobre esos campos. En esta ocasión se utilizaron solamente las bibliotecas LangDetect, Polyglot y Langid[13] con un enfoque zero-shot, es decir, no se modificaron ni se realizó un ajuste fino sobre los parámetros del modelo original de la biblioteca. Simplemente se utilizó cada uno de ellos para predecir el idioma de los textos sin necesidad de un ajuste fino adicional para el conjunto de datos.

[12] Aplicamos aquí el término zero-shot entendido como cualquier enfoque que permita aplicar un modelo preentrenado a una tarea o clase que no fue explícitamente incluida durante su entrenamiento, sin necesidad de reentrenamiento ni ajuste fino (fine-tuning). Aunque el término suele asociarse a modelos de lenguaje generativos y al uso de prompting, su definición se extiende a modelos discriminativos como langid, langdetect o fastText cuando se utilizan directamente sobre nuevos textos o dominios sin adaptación previa.

[13] En esta ocasión se descartaron las bibliotecas CLD3 y Textcat. La primera de ellas presentaba, curiosamente porque se trata de la misma empresa de desarrollo, ciertas incompatibilidades con Google Colab. En el segundo caso, porque requería la instalación y ejecución en R y se prefirió simplificar la tarea.

LangDetect

La biblioteca LangDetect[14] es una herramienta basada en la biblioteca de Google Language Detection (Compact Language Detector 2) (Shuyo, 2010). Utiliza algoritmos de aprendizaje automático para predecir el idioma de un fragmento de texto. Tiene soporte para múltiples idiomas (más de 55). Se trata de una herramienta relativamente ligera que no requiere una gran cantidad de recursos para funcionar y ofrece resultados confiables.

Polyglot

Polyglot[15] es una biblioteca que soporta una amplia gama de tareas y lenguajes (Lui et al., 2014). Puede manejar más de 100 idiomas y posee soporte para una enorme serie de tareas de PLN (como tokenización, reconocimiento de entidades nombradas, análisis de sentimiento, traducción de palabras, etc.). Una de sus desventajas es que depende de varias bibliotecas y herramientas externas, lo que hace un poco más compleja su instalación y configuración.

Langid

Langid[16] es una herramienta de software libre y de código abierto que puede identificar entre 97 y más de 100 idiomas (Lui & Baldwin, 2011). Está optimizada para ser rápida y eficiente en términos de uso de memoria y tiempo de procesamiento, inclusive en tareas de procesamiento de texto en tiempo real. Es autocontenida, no depende de servicios externos ni de bases de datos de idiomas, lo que la hace fácilmente instalable y desplegable en cualquier entorno.

NUEVO ENFOQUE: TAREAS DE LIMPIEZA Y PRE-PROCESAMIENTO DE TEXTOS

Durante las tareas realizadas previamente se observó que varios resúmenes no poseían la etiqueta de idioma y que algunos tenían diferentes formatos y caracteres que podían introducir ruido o dificultar el procesamiento (como casos con código html o LáTex o inclusive enumeraciones con viñetas que parecían confundir a las bibliotecas y modelos).

En esta ocasión, al revisar más detalladamente el dataset se logró identificar algunos patrones recurrentes en determinados errores de catalogación: existían resúmenes que contenían el mismo texto en dos idiomas diferentes en la misma celda a los que denominamos *resúmenes dobles*. Esto último era un problema generado al exportar los datos del repositorio pero que estaba ocasionado por los catalogadores humanos: cuando un catalogador había clasificado en un ítem dos resúmenes diferentes con el mismo idioma el sistema concatenaba

[14] Disponible en: https://pypi.org/project/langdetect
[15] Disponible en: https://github.com/saffsd/polyglot
[16] Disponible en: https://github.com/saffsd/langid.py

ambos resúmenes en una misma columna provocando un bloque de texto de muy difícil clasificación puesto que podría contener, por ejemplo, un 50% de texto en español y un 50% de texto en otro idioma.

Por estas razones se diseñaron varias tareas de limpieza de texto. En una primera etapa, se recorrieron las columnas que contenían resúmenes dobles y se extrajo el idioma asociado a cada uno, tomando como referencia el nombre de la columna. Luego, se aplicaron reglas específicas para clasificar los casos en tres categorías:

1. Resúmenes que indicaban explícitamente "No posee", o "No se posee" es decir, el catalogador humano había anotado explícitamente que no se tenía un resumen para ese ítem;
2. resúmenes dobles, es decir, compuestos por dos partes concatenadas (en el caso de DSpace mediante el símbolo ||), y
3. resúmenes sin un idioma declarado, identificados como sin idioma, porque correspondían a la clase "otro" utilizada para anotar idiomas diferentes al español, inglés, portugués, francés, italiano o alemán en la interfaz DSpace[17].

En una segunda etapa, se procedió a limpiar el contenido textual de los resúmenes reorganizados. La limpieza incluyó la eliminación de fragmentos de código LaTeX (*\textbf{...}*), bloques de MathML (*$...$*), etiquetas HTML (*<p>, *, etc.), viñetas o símbolos al inicio de línea, texto entre paréntesis o corchetes, URLs, direcciones de correo electrónico y espacios múltiples. También se aplicó normalización de caracteres utilizando la biblioteca *unicodedata*[18], lo cual permitió transformar letras acentuadas en sus equivalentes básicos (por ejemplo, á → a) y corregir errores comunes de codificación en archivos mal interpretados, como *â€™* por *'*.

Todas las transformaciones aplicadas durante la limpieza fueron registradas en un informe detallado, en el que se consignó el tipo de contenido eliminado o reemplazado, el identificador del ítem correspondiente y el contenido afectado. Como resultado, se generaron dos archivos: uno con los resúmenes limpios y otro con el informe completo de las modificaciones realizadas, lo que permitió mantener trazabilidad y control de calidad en el preprocesamiento del corpus.

Luego de separar los resúmenes que estaban concatenados con el símbolo ||, se identificaron 546 casos de resúmenes dobles, 1.160 resúmenes sin idioma declarado, y 27 que indicaban explícitamente que no poseían resumen. El total de registros resultantes tras la reestructuración fue de 151.532 resúmenes diferentes. Para un análisis más objetivo y correcto, se comparó la detección de las bibliotecas con respecto a los humanos quitando los casos en los

[17] Esta particularidad se debe a que en el caso de SEDICI suele ser muy poco común la aparición de otros idiomas aunque, como veremos, ocurre cada vez con mayor frecuencia.
[18] https://docs.python.org/3/library/unicodedata.html

que el idioma no estaba catalogado (ya que ese campo se encontraba vacío y obviamente se obtendría un resultado diferente), los casos en los que dos o más resúmenes estaban catalogados con el mismo idioma (ya que al menos uno de ellos estaría siempre mal catalogado en el dataset original) y los casos en los que solo se había anotado que no se poseía resumen, porque se trataba además de un texto demasiado corto y no era estrictamente un resumen.

RESULTADOS

Se evaluó el desempeño de cada biblioteca en un entorno estandarizado de Google Colab con un procesador Intel(R) Xeon(R) CPU @ 2.20GHz, con 4 núcleos físicos y 8 núcleos lógicos, 54.75 GB de memoria RAM.

Al igual que en la ocasión anterior el desempeño de las diferentes bibliotecas con las que se aplicó el enfoque zero-shot fue similar en cuanto a la coincidencia del idioma detectado respecto del idioma catalogado por los administradores humanos. Como en algunos casos las tareas de PLN pueden requerir el uso de recursos importantes, se evaluó además el tiempo requerido para el procesamiento de los datos y la detección de idiomas con una CPU y alta memoria ram (Tabla 2).

Tabla 2 - Porcentaje de coincidencias y desempeño en la detección de idiomas entre catalogadores humanos y diferentes biblitotecas. Elaboración propia.

Biblioteca	Tiempo de ejecución	Total de registros comparados	Iguales	Porcentaje Iguales
Lang Detect	13 minutos y 33.23 segundos	149253	147002	98.49%
Langid	7 minutos y 22.48 segundos	149253	147290	98.68%
Polyglot	0 minutos y 47.05 segundos	149253	143818	96.36%

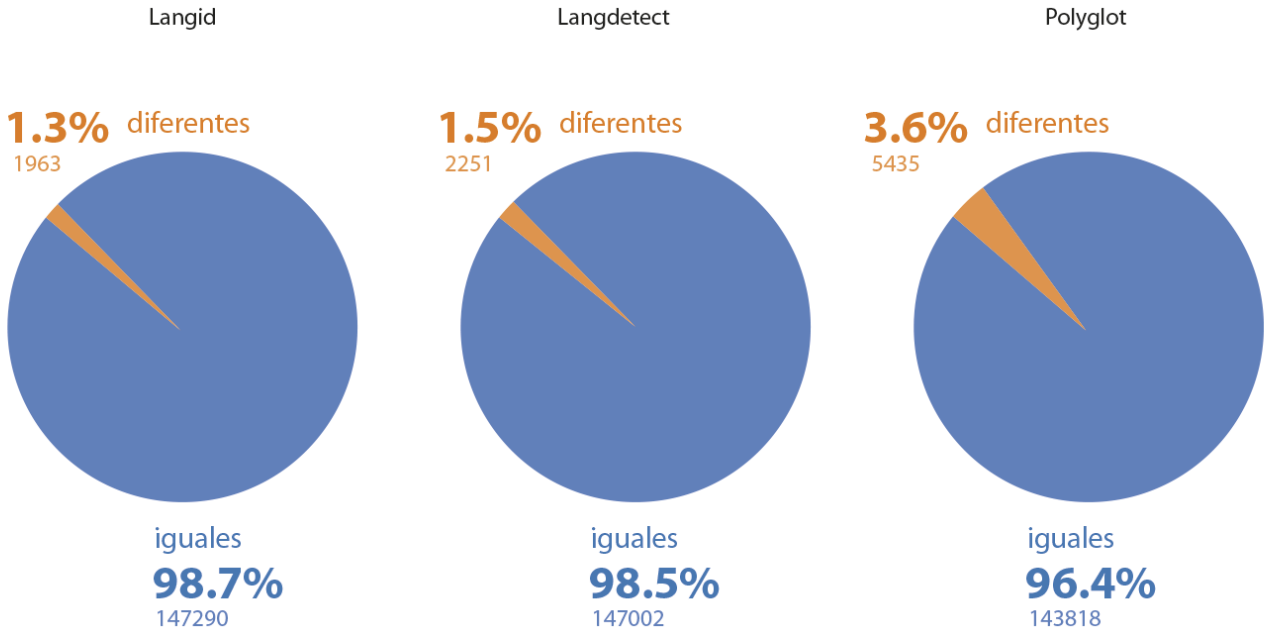
Fuente: elaborabión propria.

En esta ocasión se observaron mejoras significativas en la coincidencia con el idioma catalogado manualmente, atribuibles a los refinamientos en la limpieza y estructuración del texto de entrada. Langid destacó como la biblioteca con mayor precisión, alcanzando un 98,68 % de coincidencias exactas sobre un total de 149.253 registros, seguida de LangDetect, que obtuvo un 98,49 %. Estos resultados representan incrementos notables en comparación con pruebas anteriores. Además, ambos detectores mostraron tiempos de ejecución aceptables, siendo Langid particularmente eficiente al completar el procesamiento en 7 minutos y 22,48 segundos.

Por otro lado, Polyglot presentó el menor porcentaje de coincidencias (96,36 %), aunque continuó evidenciando una gran velocidad, ya que finalizó la tarea en tan solo 47,05 segundos. Esta característica lo posiciona también como una de las opciones óptimas en escenarios donde el tiempo de procesamiento resulte relevante.

Con estos resultados se hizo ostensible que las mejoras en el preprocesamiento textual habían contribuido de manera sustancial a un mejor desempeño global de las herramientas evaluadas (Figura 4).

Figura 4. Desempeño de reconocimiento de idiomas con enfoque *zero-shot* luego de tareas de limpieza y preprocesamiento.



Fuente: Elaboración propia.

PRECISIÓN Y GRADO DE ACUERDO ENTRE BIBLIOTECAS PARA LOS TRES SUBCONJUNTOS: RESÚMENES ORIGINALES, RESÚMENES DOBLES Y RESÚMENES SIN IDIOMA CATALOGADO

A continuación se tomaron como etiquetas válidas a todas aquellas etiquetas de idioma en las que las bibliotecas y los catalogadores humanos habían coincidido. Se revisaron las etiquetas reales de idioma en todos los casos en los que las bibliotecas y los catalogadores humanos originales habían diferido, tanto en el dataset original, en los resúmenes dobles o en los resúmenes sin idioma catalogado, que no las tenían. Esta tarea era fundamental para conocer el desempeño real de cada biblioteca evaluada así como de los catalogadores humanos. Posteriormente se comparó el grado de acuerdo en la clasificación de idiomas para LangDetect, LangID y Polyglot. La comparación se aplicó sobre tres conjuntos de resúmenes: el caso de los resúmenes dobles, el caso de los resúmenes sin idioma catalogado, y finalmente, aquellos en

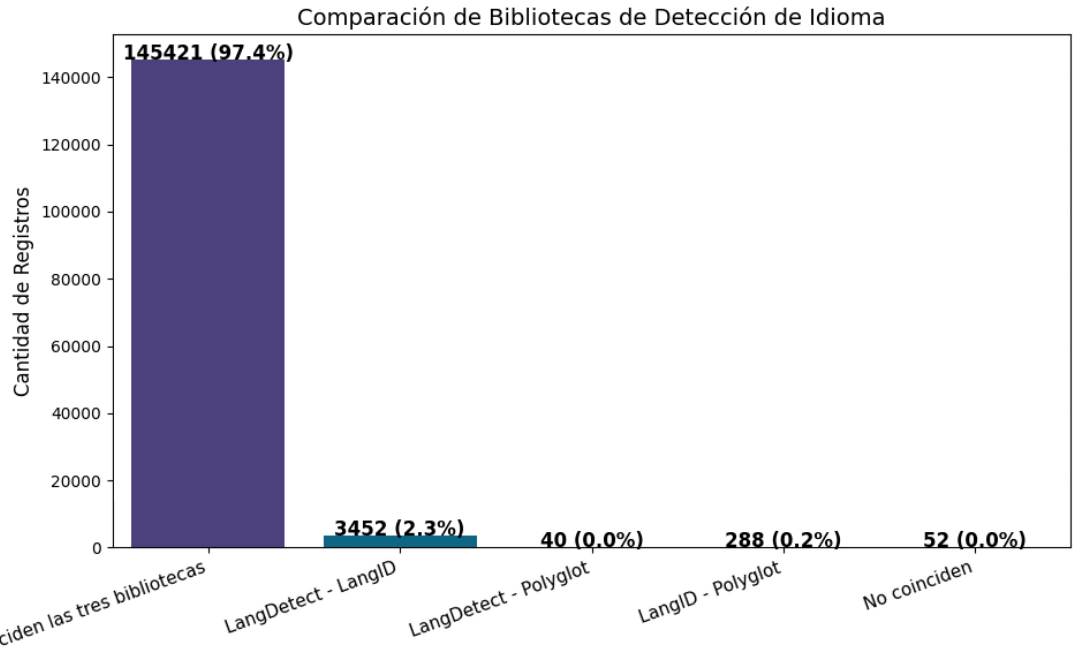
los que al menos una de las bibliotecas difería respecto del catalogador humano.

La comparación de coincidencia de bibliotecas se estructuró en cinco categorías:

- Coincidencia total: los tres sistemas coinciden en el idioma detectado.
- Coincidencia parcial (2 de 3 coinciden) con sus posibles pares (LangDetect–LangID, LangDetect–Polyglot, LangID–Polyglot).
- Desacuerdo total: los tres sistemas asignan idiomas distintos.

En el caso de los resúmenes que habían sido catalogados por los administradores originales de DSpace y tenian una etiqueta única de idioma la coincidencia de las bibliotecas fue del 97,4% (Figura 5).

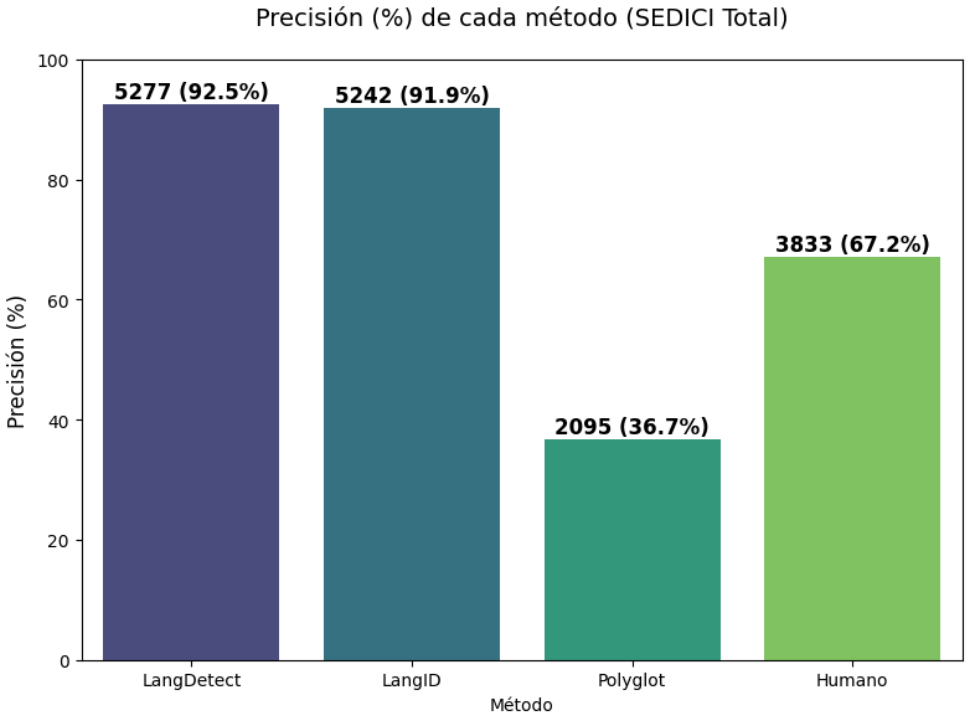
Figura 5 - Coincidencia de bibliotecas en los resúmenes que tenían una etiqueta de idioma en el *dataset* original.



Fuente: Elaboración propia.

Posteriormente se comparó el desempeño de las diferentes bibliotecas para los casos en los que al menos una de ellas había diferido con la catalogación de los administradores humanos, en este caso, la que mejor desempeño obtuvo fue LangDetect que fue capaz de asignar el idioma correcto al 92,5% de los casos mientras que Polyglot fue la de peor desempeño, debajo, inclusive del desempeño de los catalogadores humanos (Figura 6).

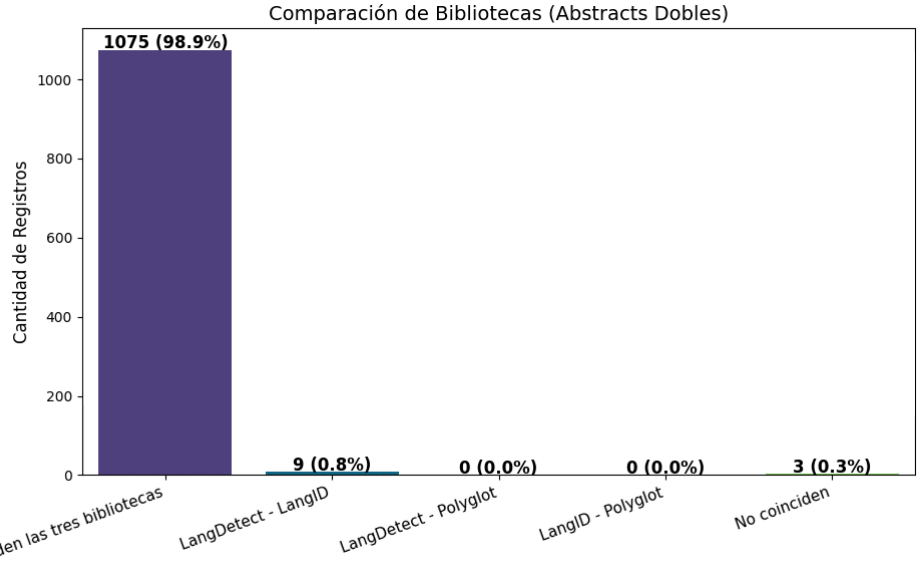
Figura 6 - Precisión en la detección de idiomas de las diferentes biblioteca y los catalogadores humanos en los casos en los que al menos una de ellas no coincidió con la catalogación original.



Fuente: Elaboración propia.

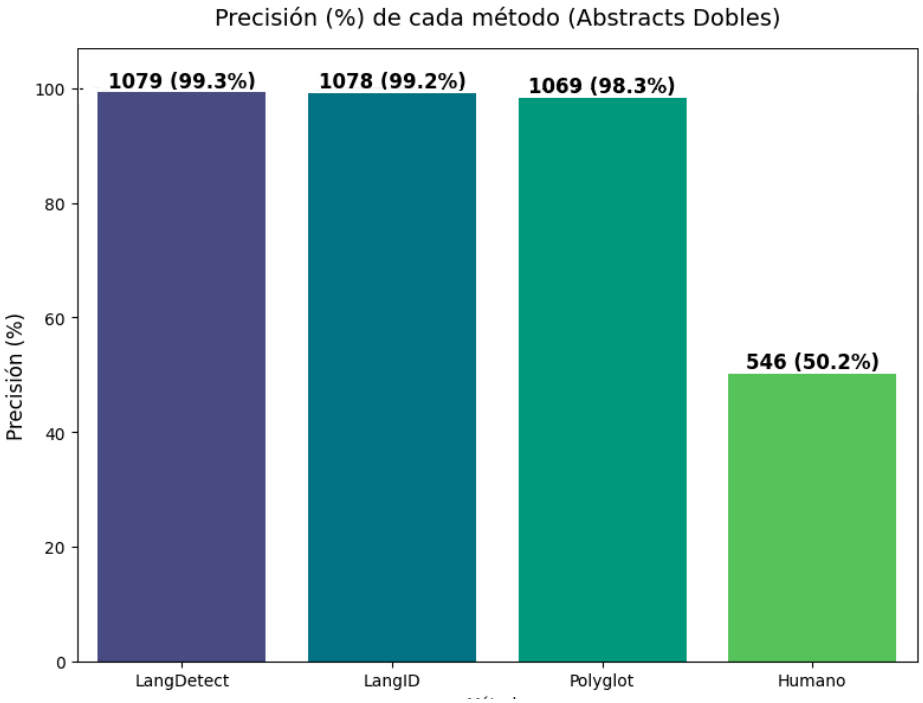
A continuación se comparó el desempeño de las diferentes bibliotecas para los resúmenes dobles y el grado de coincidencia de las mismas. En este caso es comprensible que la coincidencia con los catalogadores orignales sea de alrededor del 50% porque los resúmenes dobles presentaban ambos la misma etiqueta por un error de catalogación (Figuras 7 y 8).

Figura 7 - Coincidencia de bibliotecas en los resúmenes dobles.



Fuente: Elaboración propia.

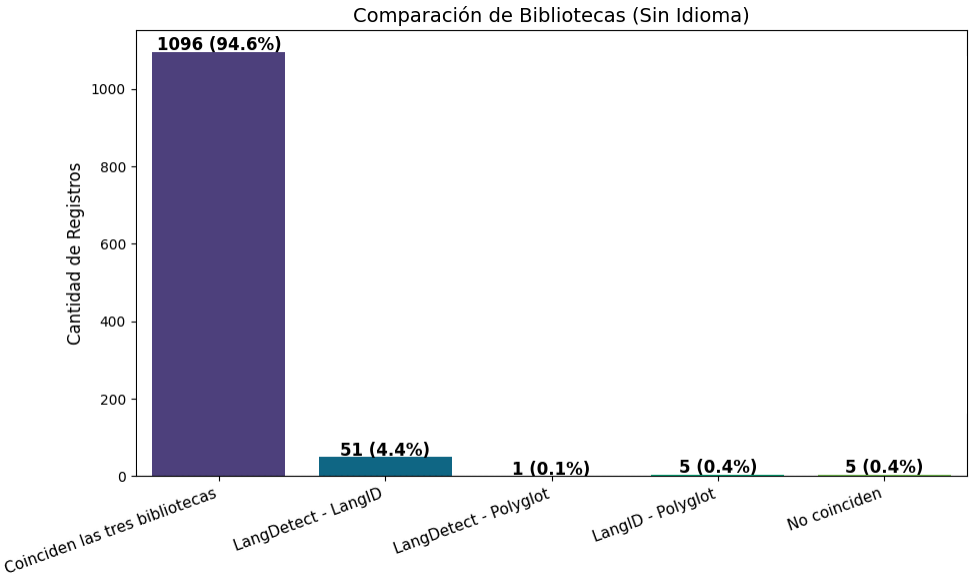
Figura 8 - Precisión de las diferentes bibliotecas y catalogadores originales en los casos de resúmenes dobles.



Fuente: Elaboración propia.

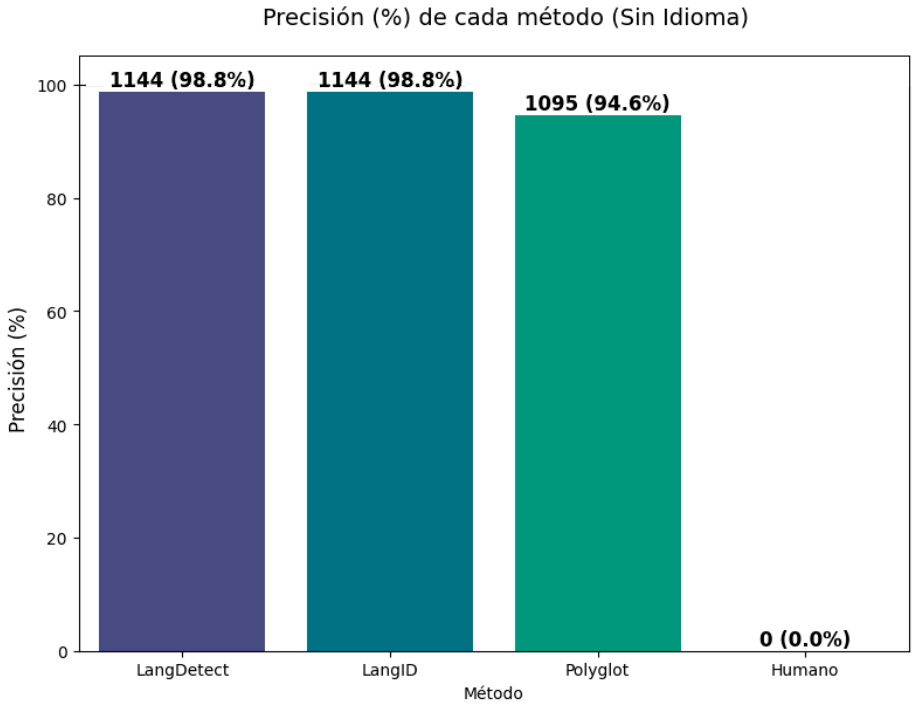
Para el caso de los resúmenes sin idioma también resulta comprensible que el catalogador humano no tenga ningún grado de precisión porque el campo idioma estaba vacío en el dataset ya que correspondía a la opción *varios*, en la interfaz de DSpace. No se trata aquí de un error de catalogación sino de la disposición de los datos en el repositorio (Figuras 9 y 10).

Figura 9 - Coincidencia de bibliotecas en los resúmenes sin idioma.



Fuente: Elaboración propia.

Figura 10 - Precisión de las diferentes bibliotecas en los casos de resúmenes dobles.



Fuente: Elaboración propia.

El corolario final de esta tarea es que en general las tres bibliotecas analizadas suelen coincidir en un alto porcentaje en los idiomas identificados y tener un alto grado de precisión. Por estas razones, cabe la posibilidad de que funcionen como un excelente insumo para la catalogación automática de los campos de idioma en el repositorio.

BIBLIOTECAS PARA LA DETECCIÓN AUTOMÁTICA DE IDIOMAS: ENFOQUE CON AJUSTE FINO

Como en la vez anterior, antes de realizar una tarea de reconocimiento de idiomas, fue necesario equilibrar las clases, es decir, los ejemplos de resúmenes con diferentes idiomas del dataset original, para poder realizar un ajuste fino de los modelos de lenguaje en la tarea con un grado mínimo de fiabilidad.

AUMENTO DE DATOS CON MARIAN MT MODEL Y M2M100

El aumento de datos es una técnica común en procesamiento del lenguaje natural (PLN) que consiste en generar ejemplos adicionales a partir de un conjunto de datos existente, manteniendo el significado del texto original. Esto se puede lograr mediante transformaciones como la sustitución de sinónimos o, en contextos multilingües, mediante traducciones automáticas. Esta estrategia busca mejorar la capacidad de generalización de los modelos cuando existen clases minoritarias con pocos ejemplos.

MODELOS UTILIZADOS PARA EL AUMENTO DE DATOS

MODELO MARIANMT

En la tarea de aumento de datos se utilizó MarianMT[19] para incrementar el número de ejemplos de las clases minoritarias (francés, portugués, italiano y alemán) a partir de traducciones de ejemplos de las clases mayoritarias (español e inglés). MarianMT forma parte de la familia de modelos de traducción automática neuronal desarrollada por el equipo de Marian NMT (Han et al., 2022; Junczys-Dowmunt et al., 2018; Tiedemann, 2012). Se trata de un modelo diseñado para ser eficiente y liviano, optimizado para aplicaciones en tiempo real y en dispositivos con recursos limitados. Es un proyecto de código abierto compatible con múltiples pares de idiomas.

MODELO M2M100

En el caso del portugués, en el que no fue posible utilizar los modelos MarianMT, se empleó como alternativa el modelo M2M100. Se trata de un modelo de traducción automática multilingüe desarrollado por Facebook AI (Fan et al., 2020), diseñado para traducir directamente entre 100 idiomas sin requerir el uso del inglés como lengua intermedia. El modelo fue entrenado con una arquitectura de tipo Transformer sobre más de 7.5 mil millones de oraciones multilingües, y es de código abierto.

En esta tarea, se realizó un aumento de datos específico sobre un conjunto de resúmenes catalogados en distintos idiomas. El archivo base contenía 149.743 ejemplos, con una distribución altamente desbalanceada: las clases mayoritarias eran español (103.791 ejemplos) e inglés (41.542 ejemplos), mientras que las clases minoritarias como francés, portugués, alemán o italiano tenían entre 13 y 3.894 ejemplos. Para evitar el desequilibrio extremo, se eliminaron del conjunto aquellas clases con menos de 2 muestras, y se mantuvieron solo los siguientes idiomas: español, inglés, portugués, francés, italiano, alemán y catalán. Posteriormente, se definió un mínimo de 4000 ejemplos para las clases minoritarias. A fin de alcanzar este objetivo, se utilizó una estrategia de traducción automática con los modelos MarianMT y el modelo M2M100 de Facebook (este último, como alternativa en caso de error de los modelos MarianMT).

- Las traducciones se generaron de la siguiente manera:
- De textos en español hacia: portugués (pt), francés (fr), italiano (it) y catalán (ca)
- De textos en inglés hacia: alemán (de)

Para cada idioma minoritario, se seleccionaron aleatoriamente textos de la clase fuente y se tradujeron al idioma objetivo. Cada texto traducido fue etiquetado con el idioma corres-

[19] Disponible en: https://huggingface.co/docs/transformers/model_doc/marian

pondiente y agregado como nuevo ejemplo en el dataset. En todos los casos se utilizaron modelos MarianMT salvo para la traducción al portugués en el que se recurrió al M2M100 por un error del modelo MarinMT que provocaba traducciones fallidas.

El conjunto de datos aumentado fue dividido en dos mitades estratificadas por idioma. La división se realizó de modo que cada clase (idioma) quedara representada proporcionalmente en ambas mitades. Para esto, se ordenaron aleatoriamente los ejemplos de cada clase y se dividieron en partes iguales, asignando una instancia adicional a la primera mitad en caso de cantidad impar. La primera mitad fue utilizada para generar los subconjuntos de entrenamiento, validación y prueba. Esta división secundaria también fue estratificada, asignando el 70 % de los ejemplos al entrenamiento, y dividiendo el 30 % restante en partes iguales para validación (15 %) y testeo (15 %). La segunda mitad, por su parte, fue reservada para futuras tareas de evaluación externa o como conjunto independiente para experimentos posteriores.

Tabla 3 - Comparación de la distribución de idiomas del dataset original y las nuevas distribuciones generadas con Marian MT Model.

Idioma	Conteo_Original	Objetivo_Target	Nuevos_Ejemplos_Generados	Modelo utilizado
es	103791	103791	0	
en	41542	41542	0	
pt	3894	4000	106	M2M100
fr	340	4000	3660	MarianMT opus-mt-es-fr
it	89	4000	3911	MarianMT opus-mt-es-it
de	62	4000	3938	MarianMT opus-mt-en-de
ca	13	4000	3987	MarianMT opus-mt-es-ca

Fuente: Elaboración propia.

MODELOS UTILIZADOS PARA LA DETECCIÓN DE IDIOMAS

En esta ocasión, se decidió darle una nueva oportunidad a FastText realizando un ajuste fino específicamente con el dataset de SEDICI, a fin de evaluar su rendimiento en condiciones controladas y comparables. Además de mBERT, se incorporaron al análisis modelos XLM-RoBERTa y SBERT multilingüe, que también fueron ajustados para la misma tarea.

FASTTEXT

FastText[20] es una biblioteca de aprendizaje automático desarrollada por Facebook AI Research (FAIR) diseñada para la clasificación de textos y la representación de palabras (Bojanowski et al., 2017; Joulin *et al.*, 2016a; Joulin et al., 2016b; Mannes, 2016, 2017). Utiliza modelos de redes neuronales para comprender la representación de las palabras en grandes conjuntos de datos de texto. Una de sus características más sobresalientes es el tratamiento de las palabras como n-gramas de caracteres por lo que puede capturar mejor el significado de palabras cortas, prefijos y sufijos, sobre todo con idiomas de morfología más rica y versátil. Posee una alta precisión en la detección de idiomas, incluso en muestras cortas.

FastText puede ser menos efectivo para algunas tareas de PLN avanzadas comparado con modelos de PLN basados en transformers, como BERT (Devlin et al., 2019), sin embargo suele desempeñarse muy eficientemente en tareas de detección de idiomas.

MODELO MBERT

mBERT[21], o multilingual BERT, es una variante del modelo BERT (Bidirectional Encoder Representations from Transformers) diseñado por Google (Devlin et al., 2019). BERT marcó un hito en el área de procesamiento del lenguaje natural (NLP) por su capacidad para comprender mejor el contexto de las palabras en un texto, comparado con los modelos anteriores. mBERT está pre entrenado en los textos de Wikipedia de 104 idiomas y es capaz procesar y entender múltiples idiomas sin necesidad de entrenamiento específico del idioma. Al utilizar tecnología de transformers requiere una cantidad de recursos computacionales considerable.

MODELO SBERT (SENTENCE-BERT)

SBERT[22], o Sentence-BERT, es un modelo basado en BERT diseñado específicamente para producir representaciones vectoriales significativas a nivel de oración especialmente en tareas como búsqueda semántica, comparación de similitud o clustering (Reimers & Gurevych, 2019). Sentence-BERT permite una comparación más eficiente de sentencias mediante el uso de redes siamesas, lo que reduce significativamente el tiempo de cómputo requerido en tareas de emparejamiento semántico.

MODELO XLM-ROBERTA

XLM-RoBERTa[23] (Cross-lingual RoBERTa) es una versión multilingüe del modelo RoBERTa (Robustly Optimized BERT Pretraining Approach), desarrollado por Facebook AI (Conneau

[20] Disponible en: https://fasttext.cc/
[21] Disponible en: https://github.com/google-research/bert/blob/master/multilingual.md
[22] Disponible en: https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1
[23] Disponible en: https://huggingface.co/FacebookAI/xlm-roberta-base

et al., 2020). Está preentrenado sobre un corpus significativamente más grande y diverso que mBERT, utilizando datos de CommonCrawl[24] en 100 idiomas lo que le brinda una mayor capacidad para representar lenguas con menos recursos y capturar matices complejos A diferencia de mBERT, que fue entrenado en Wikipedia, XLM-RoBERTa posee un vocabulario y arquitectura ajustados para mejorar el desempeño en tareas multilingües, incluyendo clasificación, detección de idioma y análisis de sentimiento.

RESULTADOS DEL MODELO FASTTEXT

En el caso del modelo FastText, en esta ocasión, se realizó el ajuste fino en una infraestructura de alto rendimiento que permitió acelerar significativamente el proceso. El entorno contaba con un procesador Intel(R) Xeon(R) CPU @ 2.20GHz, con 6 núcleos físicos y 12 núcleos lógicos, acompañado por una memoria RAM de 89,63 GB. Además, el sistema contaba con una GPU NVIDIA A100-SXM4-40GB, aunque el ajuste fino de FastText no hace uso directo de GPU, ya que se trata de un modelo optimizado para ejecución eficiente en CPU. La misma configuración de hardware se utilizó para el resto de los modelos.

FastText fue ajustado con los siguientes parámetros: 5 épocas, un *learning rate* de 1 y con la modalidad de bigramas de palabras[25] y fue evaluado sobre un conjunto de prueba estratificado que incluyó siete clases de idioma: español, inglés, portugués, francés, italiano, alemán y catalán. Los resultados obtenidos fueron altamente satisfactorios: la precisión promedio ponderada fue del 100 por ciento, y el valor de *f1-score* se mantuvo entre 0.99 y 1.00 en todas las clases. En particular, las clases mayoritarias como español e inglés lograron un desempeño casi perfecto, y las clases minoritarias, incluidas aquellas generadas por aumento de datos como catalán, francés o alemán, mostraron una excelente capacidad de generalización. El tiempo total de ajuste fino y evaluación fue de apenas 13.58 segundos.

Tabla 4. Reporte de Clasificación FastText.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,98	0,99	300
de	1	1	1	300
en	1	0,99	0,99	3116

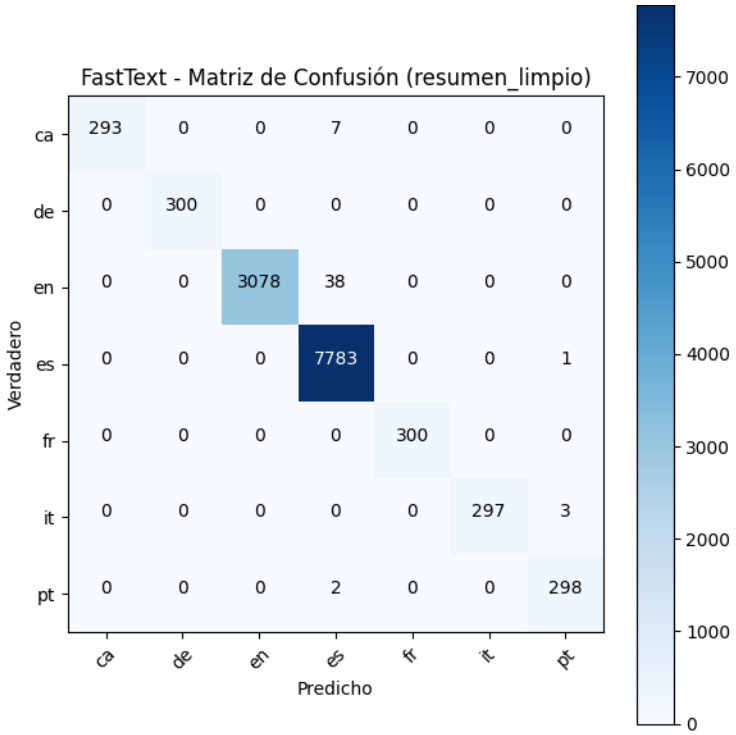
[24] CommonCrawl es un proyecto que recopila y publica mensualmente datos abiertos de miles de millones de páginas web en distintos idiomas, utilizado frecuentemente para entrenar modelos de lenguaje multilingües. Más información en commoncrawl.org.

[25] El parámetro epoch=5 indica que el modelo recorrió todo el conjunto de entrenamiento cinco veces durante el proceso de ajuste. El valor lr=1.0 corresponde a la tasa de aprendizaje, que controla la magnitud de las actualizaciones de los pesos internos del modelo; un valor alto como este acelera el aprendizaje inicial. Finalmente, wordNgrams=2 habilita el uso de bigramas, es decir, secuencias de dos palabras, lo que permite al modelo capturar cierta información contextual más allá de palabras individuales.

Idioma	Precision	Recall	F1-Score	Soporte
es	0,99	1	1	7784
fr	1	1	1	300
it	1	0,99	0,99	300
pt	0,99	0,99	0,99	300
accuracy	1	1	1	12400
macro avg	1	0,99	0,99	12400
weighted avg	1	1	1	12400

Fuente: Elaboración propia.

Figura 11 - Matriz de confusion FastText.



Fuente: Elaboración propia.

EVALUACIÓN SOBRE LA SEGUNDA MITAD DEL DATASET

En una segunda fase de evaluación, se utilizó el modelo FastText previamente ajustado para predecir el idioma de los textos correspondientes a la segunda mitad del conjunto de datos aumentado. Este conjunto incluyó un total de 82.666 ejemplos distribuidos entre los siete idiomas anteriores. El modelo alcanzó un desempeño sobresaliente, con una precisión global

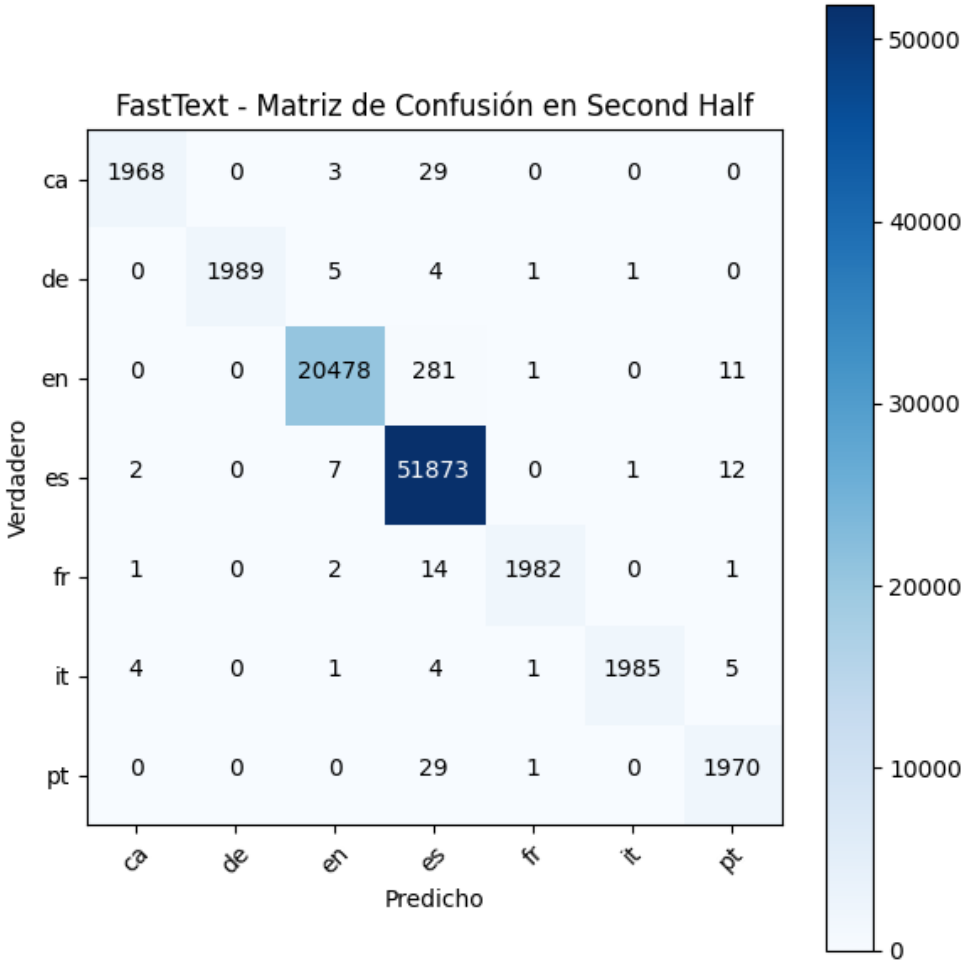
(accuracy) del 99 %. Los valores de *precision*, *recall* y *f1-score* fueron igualmente altos para todas las clases. Se observaron valores de f1-score iguales o superiores a 0.99 en todas las lenguas, una excelente capacidad del modelo para clasificar correctamente textos inclusive en lenguas que inicialmente presentaban una baja representación. El tiempo total requerido para ejecutar la carga del modelo, procesar los textos y generar las predicciones fue de 11.69 segundos.

Tabla 5 - Reporte de Clasificación FastText con segunda mitad del dataset.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,98	0,99	2000
de	1	0,99	1	2000
en	1	0,99	0,99	20771
es	0,99	1	1	51895
fr	1	0,99	0,99	2000
it	1	0,99	1	2000
pt	0,99	0,98	0,99	2000
accuracy	0,99	0,99	0,99	82666
macro avg	1	0,99	0,99	82666
weighted avg	0,99	0,99	0,99	82666

Fuente: Elaboración propia.

Figura 12 - Matriz de confusion FastText con segunda mitad del dataset original aumentado.



Fuente: Elaboración propia.

RESULTADOS DE MBERT

AJUSTE FINO DEL MODELO MBERT CON PÉRDIDA PONDERADA

El ajuste fino de este modelo se realizó sobre el corpus balanceado a partir de la primera mitad del dataset aumentado, aplicando una estrategia de pérdida ponderada para compensar aún más el desbalance de clases. El modelo se ajustó usando la subclase *CustomTrainer* que permite incorporar una función de pérdida personalizada (*CrossEntropyLoss*) con pesos por clase definidos de acuerdo con la frecuencia relativa en el conjunto de entrenamiento.

Los pesos por clase utilizados fueron para cada etiqueta fueron:

- Catalán: 5.90,
- Alemán: 5.90,
- Inglés: 0.57,

- Español: 0.23,
- Fracéss: 5.90,
- Italiano: 5.90,
- Portugués: 5.90.

El modelo fue ajustado durante cinco épocas, aunque se observó que el rendimiento óptimo se alcanzó al finalizar la primera. A partir de la segunda época, se registró un aumento constante en la función de pérdida sobre el conjunto de validación, lo que indicaba *overfitting*. El tiempo total de ajuste fino fue de 34 minutos y 51 segundos.

Métricas de evaluación

Evaluación en entrenamiento (accuracy: 0.97):

- Macro F1: 0.81
- Weighted F1: 0.96

Evaluación en validación (accuracy: 0.97):

- Macro F1: 0.80
- Weighted F1: 0.96

Evaluación en test (accuracy: 0.97):

- Macro F1: 0.81
- Weighted F1: 0.96

Las clases mayoritarias (‘es’ y ‘en’) obtuvieron valores de F1 superiores a 0.99, mientras que la clase ‘it’ no fue correctamente aprendida con valores de 0.00 en todas las métricas.

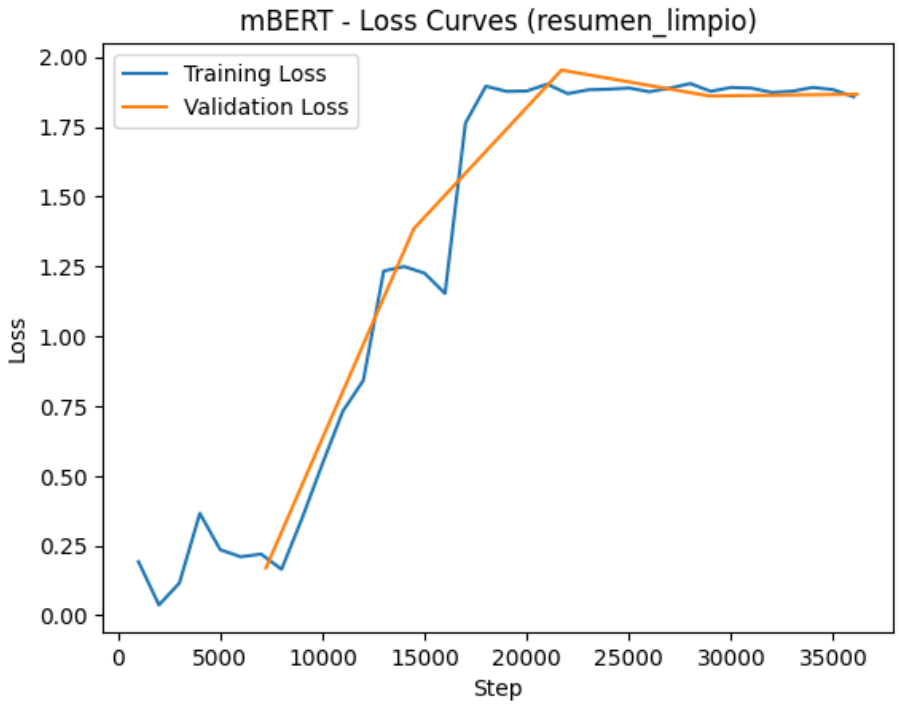
Tabla 6 - Pérdida de entrenamiento y validación y precisión del modelo mBERT en cada una de las épocas.

Época	Training Loss	Validation Loss	Accuracy
1	0,2197	0,170289	0,970164
2	1,2496	1,384677	0,898637
3	1,9028	1,953063	0,25127

Época	Training Loss	Validation Loss	Accuracy
4	1,905	1,86048	0,627772
5	1,8577	1,866561	0,627772

Fuente: Elaboración propia.

Figura 13 - Curva Loss del modelo mBERT durante el entrenamiento y la validación.



Fuente: Elaboración propia.

Tabla 7 - Reporte de Clasificación de Entrenamiento modelo mBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	1400
de	0,5	0,99	0,66	1400
en	1	0,99	0,99	14539
es	0,99	1	1	36327
fr	1	1	1	1400
it	0	0	0	1400
pt	0,99	0,99	0,99	1400
accuracy	0,97	0,97	0,97	57866
macro avg	0,78	0,85	0,81	57866
weighted avg	0,96	0,97	0,96	57866

Fuente: Elaboración propia.

Tabla 8 - Reporte de Clasificación de Validación modelo mBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	300
de	0,5	1	0,67	300
en	1	0,98	0,99	3116
es	0,99	1	1	7785
fr	1	1	1	300
it	0	0	0	300
pt	0,99	0,98	0,98	300
accuracy	0,97	0,97	0,97	12401
macro avg	0,78	0,85	0,8	12401
weighted avg	0,96	0,97	0,96	12401

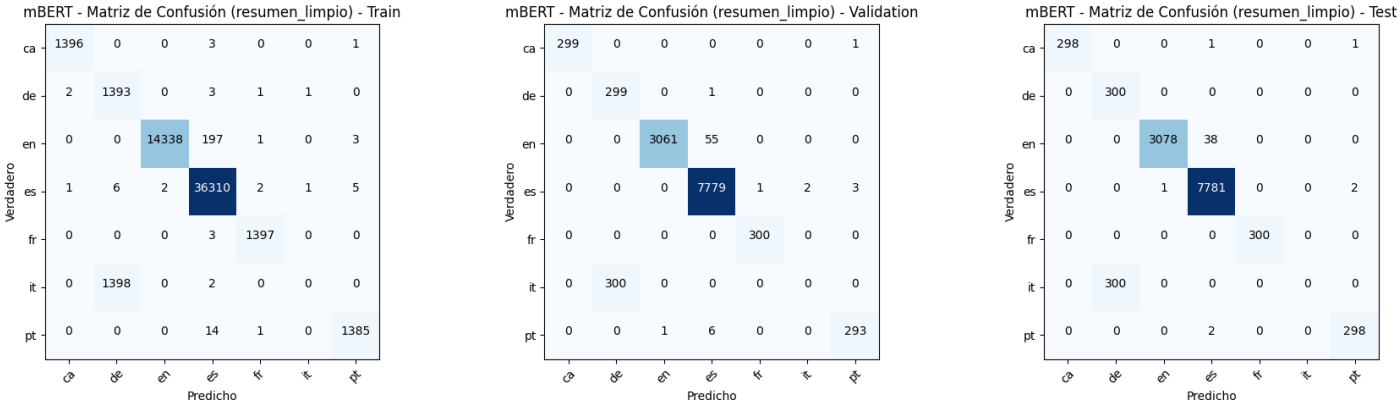
Fuente: Elaboración propia.

Tabla 9 - Reporte de Clasificación de Testeo modelo mBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	1	300
de	0,5	1	0,67	300
en	1	0,99	0,99	3116
es	0,99	1	1	7784
fr	1	1	1	300
it	0	0	0	300
pt	0,99	0,99	0,99	300
accuracy	0,97	0,97	0,97	12400
macro avg	0,78	0,85	0,81	12400
weighted avg	0,96	0,97	0,96	12400

Fuente: Elaboración propia.

Figura 14 - Matrices de confusión para las etapas de Entrenamiento, Validación y Testeo del modelo mBERT.



Fuente: Elaboración propia.

EVALUACIÓN SOBRE LA SEGUNDA MITAD DEL DATASET ORIGINAL AUMENTADO

Para evaluar la robustez del modelo más allá de los datos vistos durante el ajuste fino, se aplicó el modelo sobre la otra mitad del corpus original, no utilizada durante el entrenamiento ni validación. El tiempo total de ejecución de detección fue de 12 minutos y 48 segundos.

Métricas de evaluación en segunda mitad

Accuracy global: 0.97

Reporte de clasificación:

- Macro F1: 0.80
- Weighted F1: 0.96[26]

Las clases mayoritarias (‘es’ y ‘en’) mantuvieron valores de F1 superiores a 0.99. Al igual que en la evaluación anterior, la clase ‘it’ no fue correctamente identificada por el modelo (F1 = 0.00).

Tabla 10 - Reporte de Clasificación de segunda mitad del dataset modelo mBERT.

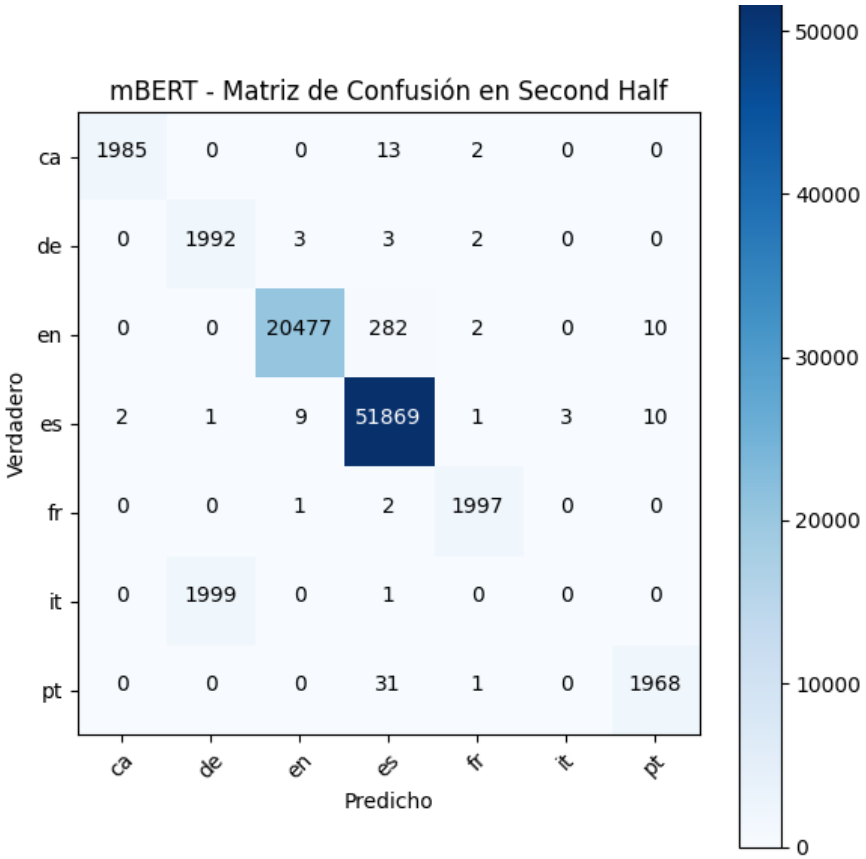
Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	1	2000
de	0,5	1	0,66	2000

[26] El F1-score macro y el F1-score ponderado (weighted) son métricas utilizadas para evaluar el rendimiento en tareas de clasificación multiclase. El macro F1 calcula el promedio del F1-score de cada clase sin tener en cuenta la cantidad de ejemplos por clase, por lo que refleja cuán equilibrado es el desempeño del modelo entre todas las clases. En cambio, el weighted F1 también promedia los F1-score de cada clase, pero ponderando según la frecuencia de cada una en el conjunto de datos, lo que lo hace más representativo del rendimiento general en presencia de clases desbalanceadas.

Idioma	Precision	Recall	F1-Score	Soporte
en	1	0,99	0,99	20771
es	0,99	1	1	51895
fr	1	1	1	2000
it	0	0	0	2000
pt	0,99	0,98	0,99	2000
accuracy	0,97	0,97	0,97	82666
macro avg	0,78	0,85	0,8	82666
weighted avg	0,96	0,97	0,96	82666

Fuente: Elaboración propia.

Figura 15 - Matriz de confusión del modelo mBERT con la segunda mitad del dataset original aumentado.



Fuente: Elaboración propia.

RESULTADOS SBERT

El modelo de clasificación basado en la arquitectura SBERT multilingüe (distiluse-base--multilingual-cased-v1) se aplicó sin pérdida ponderada. El ajuste fino se realizó con precisión

mixta (fp16)[27] durante 6 épocas, con una política de evaluación y guardado al final de cada una para luego utilizar el modelo en su estado de mejor desempeño. La mejor época fue la tercera, con una *eval_loss* de 0.01435 y una precisión del 99.48 % en validación. El ajuste completo tomó 25 minutos y 9 segundos. Los resultados mostraron un desempeño robusto en los tres conjuntos principales:

- Entrenamiento: *accuracy* = 100.0 %, *f1-score* macro = 1.00
- Validación: *accuracy* = 99.49 %, *f1-score* macro = 0.995
- Test: *accuracy* = 99.97 %, *f1-score* macro = 0.995

Los errores fueron escasos y se concentraron en algunas confusiones mínimas entre el portugués y el español, o el inglés y el español.

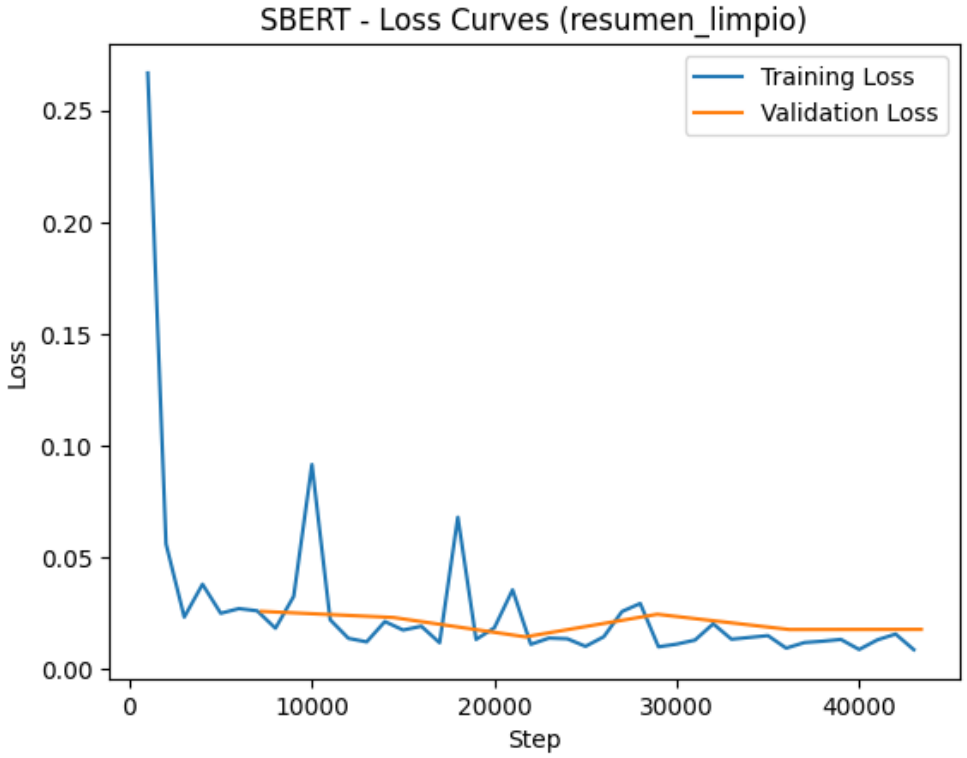
Tabla 11 - Pérdida de entrenamiento y validación y precisión del modelo SBERT en cada una de las épocas.

Época	Training Loss	Validation Loss	Accuracy
1	0,026	0,025769	0,993791
2	0,0212	0,023019	0,994355
3	0,0354	0,014354	0,994839
4	0,0293	0,024468	0,994678
5	0,0093	0,017632	0,99492
6	0,0086	0,01769	0,994758

Fuente: Elaboración propia.

[27] La precisión mixta (fp16) es una técnica de ajuste fino en la que se combinan operaciones en precisión de 16 bits (float16) y 32 bits (float32). Su uso permite reducir el consumo de memoria y acelerar el proceso de ajuste fino en GPU, manteniendo la estabilidad numérica y la precisión del modelo.

Figura 16 - Curva Loss del modelo SBERT durante el entrenamiento y la validación.



Fuente: Elaboración propia.

Tabla 12 - Reporte de Clasificación de Entrenamiento modelo SBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	1400
de	1	1	1	1400
en	1	0,99	0,99	14539
es	0,99	1	1	36327
fr	1	1	1	1400
it	1	1	1	1400
pt	1	0,99	0,99	1400
accuracy	1	1	1	57866
macro avg	1	1	1	57866
weighted avg	1	1	1	57866

Fuente: Elaboración propia.

Tabla 13 - Reporte de Clasificación de Validación modelo SBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	300
de	1	1	1	300
en	1	0,98	0,99	3116
es	0,99	1	1	7785
fr	1	1	1	300
it	1	1	1	300
pt	1	0,98	0,99	300
accuracy	0,99	0,99	0,99	12401
macro avg	1	0,99	1	12401
weighted avg	0,99	0,99	0,99	12401

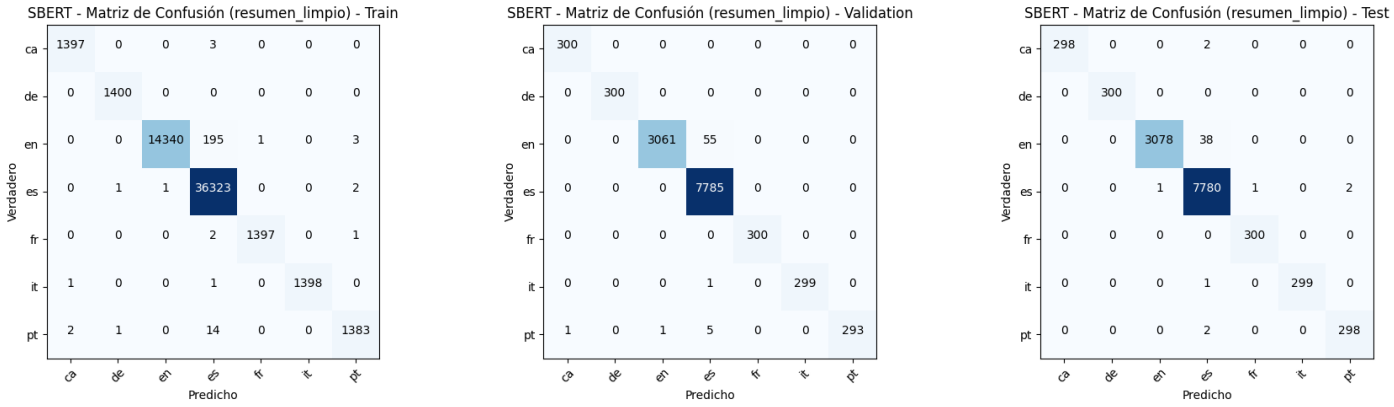
Fuente: Elaboración propia.

Tabla 14. Reporte de Clasificación de Testeo modelo SBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	1	300
de	1	1	1	300
en	1	0,99	0,99	3116
es	0,99	1	1	7784
fr	1	1	1	300
it	1	1	1	300
pt	0,99	0,99	0,99	300
accuracy	1	1	1	12400
macro avg	1	1	1	12400
weighted avg	1	1	1	12400

Fuente: Elaboración propia.

Figura 17 - Matrices de confusión para las etapas de Entrenamiento, Validación y Testeo del modelo SBERT.



Fuente: Elaboración propia.

PRUEBA EXTERNA: SEGUNDA MITAD DEL DATASET ORIGINAL AUMENTADO

Este modelo también se evaluó con la segunda mitad del dataset aumentado originalmente. El tiempo de detección completo fue de 7 minutos y 49 segundos. Los resultados fueron igualmente sólidos:

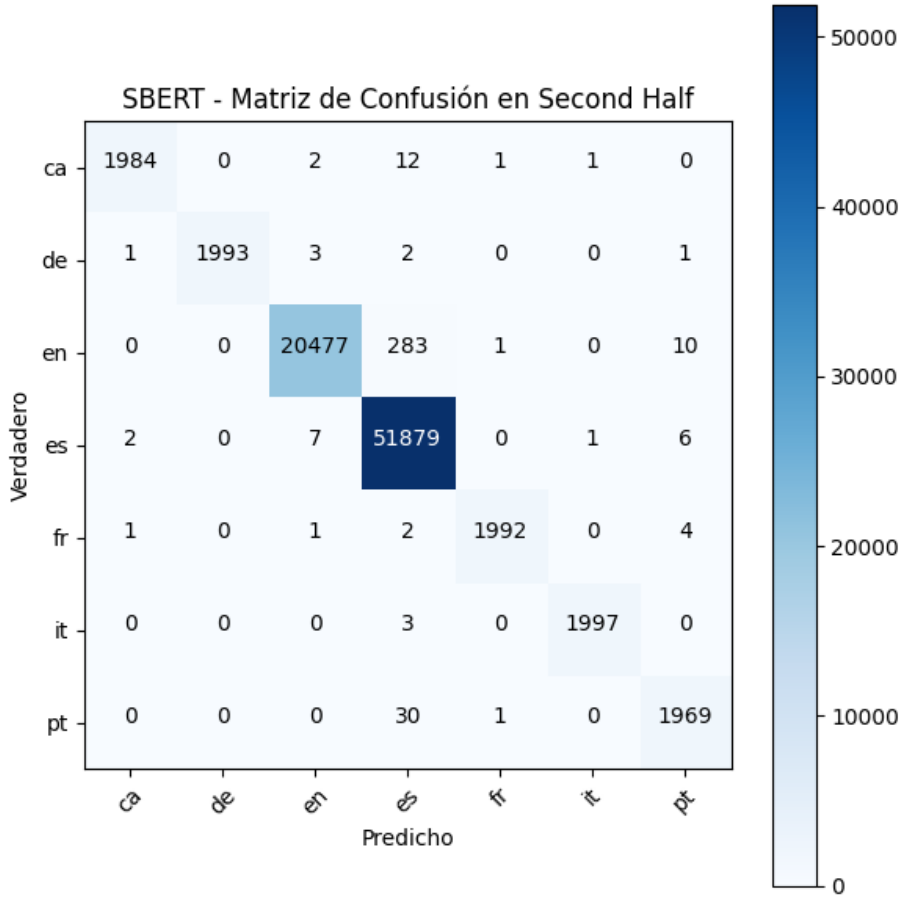
- Accuracy: 100.0 %
- f1-score macro: 0.995
- Precision/Recall por clase: todas ≥ 0.98.

Tabla 15 - Reporte de Clasificación de segunda mitad del dataset, modelo SBERT.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	0,99	2000
de	1	1	1	2000
en	1	0,99	0,99	20771
es	0,99	1	1	51895
fr	1	1	1	2000
it	1	1	1	2000
pt	0,99	0,98	0,99	2000
accuracy	1	1	1	82666
macro avg	1	0,99	1	82666
weighted avg	1	1	1	82666

Fuente: Elaboración propia.

Figura 18 - Matriz de confusión del modelo SBERT con segunda mitad del dataset original aumentado.



Fuente: Elaboración propia.

RESULTADOS XLM-ROBERTA

Se realizó un ajuste fino sobre el modelo durante 3 épocas también con precisión mixta (fp16). El proceso de ajuste tuvo una duración total de 25 minutos y 9 segundos, y la mejor época resultó ser la primera, con una pérdida de validación mínima de 0.0171. Las curvas de pérdida mostraron una evolución estable tanto en entrenamiento como en validación, sin señales de sobreajuste significativo. Los resultados obtenidos sobre los tres conjuntos fueron notablemente altos:

- Entrenamiento: accuracy = 100 %, macro-F1 = 1.00
- Validación: accuracy ≈ 99.47 %, macro-F1 ≈ 0.99
- Test: accuracy ≈ 99.95 %, macro-F1 ≈ 1.00

En el conjunto de test, todas las clases alcanzaron valores de f1-score iguales o superiores a 0.99. Se observó un rendimiento alto en clases mayoritarias como español (f1-score = 1.00) y también en clases minoritarias como catalán, francés e italiano (f1-score ≥ 0.99). Las ma-

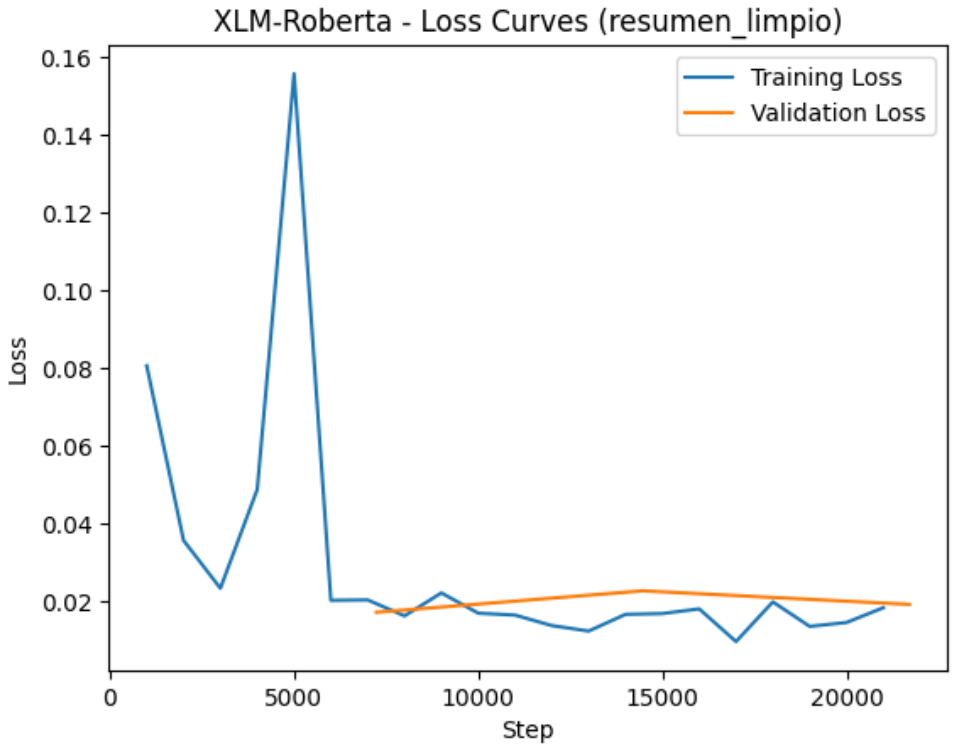
trices de confusión confirmaron una baja tasa de errores de clasificación entre idiomas afines, con un reducido número de confusiones entre inglés y portugués, o entre catalán y español. Al igual que el caso anterior de SBERT, XLM-RoBERTa logró generalizar de manera efectiva los textos incluso en presencia de clases desbalanceadas, sin necesidad de técnicas adicionales como pérdida ponderada.

Tabla 16 - Pérdida de entrenamiento y validación y precisión del modelo XLM-RoBERTa en cada una de las épocas

Época	Training Loss	Validation Loss	Accuracy
1	0,0203	0,017125	0,994678
2	0,0166	0,022643	0,994678
3	0,0183	0,019154	0,994758

Fuente: Elaboración propia.

Figura 19 - Curva Loss del modelo XLM-RoBERTa durante el entrenamiento y la validación.



Fuente: Elaboración propia.

Tabla 17 - Reporte de Clasificación de Entrenamiento modelo XLM-RoBERTa.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	1400
de	1	1	1	1400
en	1	0,99	0,99	14539
es	0,99	1	1	36327
fr	1	1	1	1400
it	1	1	1	1400
pt	1	0,99	0,99	1400
accuracy	1	1	1	57866
macro avg	1	1	1	57866
weighted avg	1	1	1	57866

Fuente: Elaboración propia.

Tabla 18 - Reporte de Clasificación de Validación modelo XLM-RoBERTa.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	1	1	300
de	1	0,99	1	300
en	1	0,98	0,99	3116
es	0,99	1	1	7785
fr	1	1	1	300
it	1	1	1	300
pt	1	0,98	0,99	300
accuracy	0,99	0,99	0,99	12401
macro avg	1	0,99	1	12401
weighted avg	0,99	0,99	0,99	12401

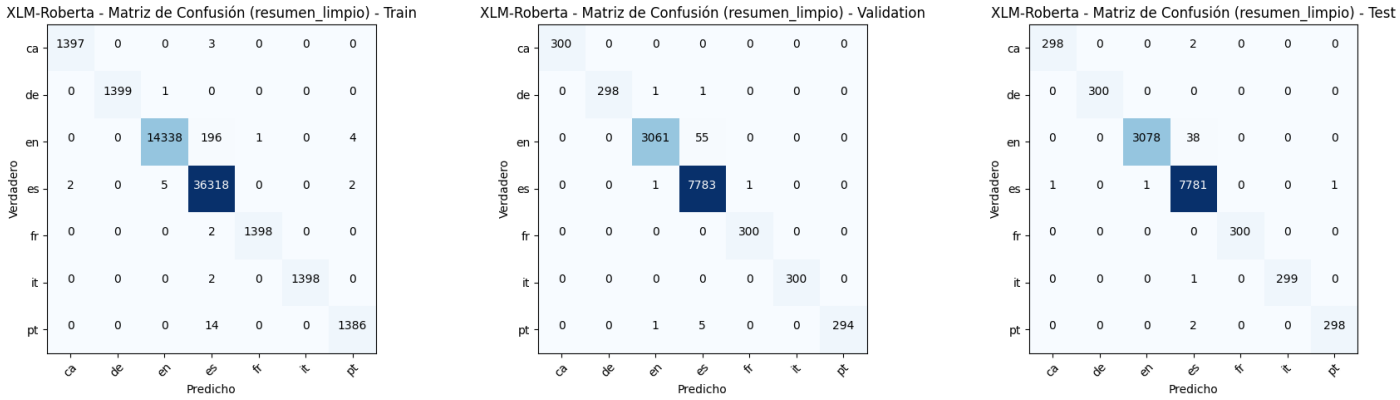
Fuente: Elaboración propia.

Tabla 19 - Reporte de Clasificación de Testeo modelo XLM-RoBERTa.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	0,99	300
de	1	1	1	300
en	1	0,99	0,99	3116
es	0,99	1	1	7784
fr	1	1	1	300
it	1	1	1	300
pt	1	0,99	0,99	300
accuracy	1	1	1	12400
macro avg	1	1	1	12400
weighted avg	1	1	1	12400

Fuente: Elaboración propia.

Figura 20 - Matrices de confusión para las etapas de Entrenamiento, Validación y Testeo del modelo XLM-RoBERTa.



Fuente: Elaboración propia.

PRUEBA EXTERNA: SEGUNDA MITAD DEL DATASET ORIGINAL AUMENTADO

También se buscó comprobar la capacidad de generalización del modelo sobre datos nuevos, representativos del dominio del corpus completo. El modelo alcanzó un rendimiento sobresaliente sobre este nuevo conjunto, con una accuracy global del 100 % y valores de f1-score iguales o superiores a 0.99 para todas las clases.

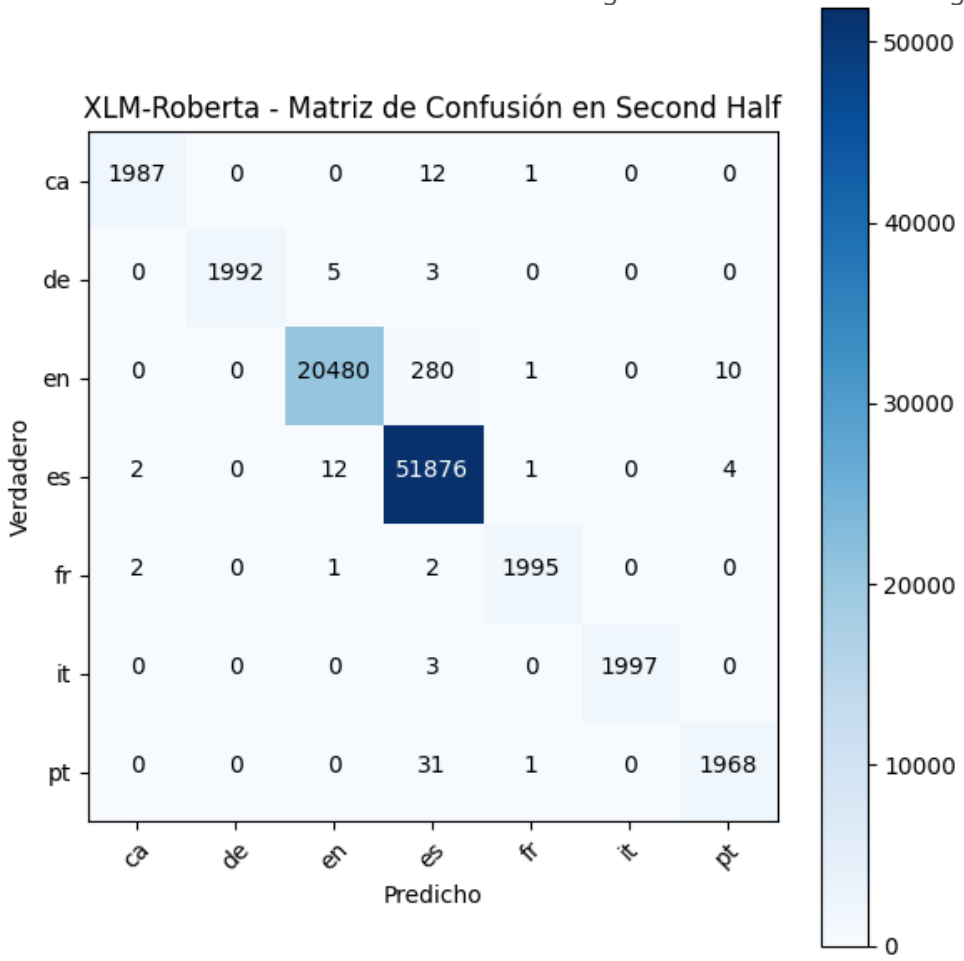
La matriz de confusión mostró una clasificación prácticamente perfecta, con muy pocos errores. El tiempo total de ejecución para procesar este conjunto fue de 14 minutos y 34 segundos.

Tabla 20 - Reporte de Clasificación de segunda mitad del dataset modelo XLM-RoBERTa.

Idioma	Precision	Recall	F1-Score	Soporte
ca	1	0,99	1	2000
de	1	1	1	2000
en	1	0,99	0,99	20771
es	0,99	1	1	51895
fr	1	1	1	2000
it	1	1	1	2000
pt	0,99	0,98	0,99	2000
accuracy	1	1	1	82666
macro avg	1	0,99	1	82666
weighted avg	1	1	1	82666

Fuente: Elaboración propia.

Figura 21 - Matriz de confusion del modelo XLM-RoBERTa con segunda mitad del dataset original aumentado.



Fuente: Elaboración propia.

DISCUSIÓN

Los resultados obtenidos en este trabajo muestran posibles enfoques para la detección automática de idiomas en repositorios institucionales. En primer lugar, se debe destacar que incluso enfoques zero-shot acompañados por un cuidadoso preprocesamiento del texto pueden alcanzar altos niveles de precisión, como fue el caso de Langid y LangDetect. En segundo lugar, respecto al tiempo de procesamiento, Polyglot, aunque con menor precisión, destacó como una excelente opción.

En cuanto a los modelos sobre los que se realizó un ajuste fino, tanto FastText como SBERT y XLM-RoBERTa, baste decir que superaron ampliamente las expectativas. No solo igualaron sino que en muchos casos superaron el rendimiento de las bibliotecas de uso general, alcanzando valores de F1 cercanos al 100 %. El uso de técnicas de aumento de datos y balanceo de clases fue fundamental para mejorar el rendimiento, especialmente en idiomas minoritarios que inicialmente estaban subrepresentados. Se destaca el caso de FastText, que con un tiempo de ajuste fino mínimo logró resultados comparables a modelos basados en BERT.

En cuanto a mBERT, si bien el modelo fue capaz de generalizar con bastante éxito en idiomas mayoritarios, presentó dificultades persistentes con clases minoritarias como el italiano, a pesar de la aplicación de pérdida ponderada. La capacidad multilingüe de mBERT no garantiza, por sí sola, una clasificación balanceada si el ajuste fino no es cuidadosamente supervisado, futuras tareas de mejora serán necesarias.

Por último, desde una perspectiva de curaduría de datos, el trabajo permitió identificar errores frecuentes en la catalogación manual —como etiquetas por defecto o resúmenes con más de un idioma concatenado— y proponer un conjunto de transformaciones trazables que mejoran significativamente la calidad del dataset (alrededor de 3000 etiquetas de resúmenes podrán ser corregidas automáticamente).

CONCLUSIONES

Este estudio ha demostrado que es técnicamente viable y metodológicamente ventajoso realizar un ajuste fino sobre modelos de lenguaje específicos para la detección automática de idiomas en textos breves, como los resúmenes de repositorios institucionales. El uso de diferentes bibliotecas y modelos no solo puede permitir la automatización de la curaduría del campo *idioma* con alta precisión, sino también contribuir con un conjunto de buenas prácticas que podrían ser replicadas en otros repositorios o bases de datos multilingües. Además, los hallazgos sobre errores humanos recurrentes aportan evidencia útil para el diseño de interfaces más robustas o validadores automáticos que reduzcan la tasa de errores en el flujo de catalogación.

De la comparación de diferentes enfoques, se puede concluir que si bien los modelos de lenguaje alcanzan una precisión superior, particularmente cuando se aplican técnicas de limpieza, aumento y balanceo de datos; dado el alto costo en cuanto a recursos de hardware necesarios para el ajuste fino de dichos modelos, el empleo de bibliotecas zero-shot que ofrecen excelentes resultados con bajo costo computacional deviene una opción mucho más conveniente y plausible.

Entre los modelos analizados, FastText se presenta como un candidato especialmente prometedor, ya que combina rapidez, precisión y facilidad de realización del ajuste fino, todo ello con un costo computacional considerablemente bajo. Esto lo convierte en una opción intermedia de alto valor para implementaciones personalizadas.

En trabajos futuros podría buscarse mejorar el desempeño del modelo mBERT o utilizar modelos aún más recientes como LaBSE (Feng et al., 2022). Una consecuencia esperable de estas tareas exploratorias y evaluatorias podría ser la incorporación de una metodología de detección y etiquetado automático de los campos de idiomas en los ítems de los repositorios. Quizá el uso combinado de bibliotecas con el enfoque zero-shot que incluya una probabilidad en la predicción del idioma o un sistema de votación podría resultar conveniente, dejando la intervención del catalogador humano para los casos en los que las bibliotecas no concuerdan.

BIBLIOGRAFÍA

Adebara, I., Elmadany, A., Abdul-Mageed, M., & Inciarte, A. A. (2023). SERENGETI: Massively Multilingual Language Models for Africa. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 1498-1537). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-acl.97

Adebara, I., Elmadany, A., Abdul-Mageed, M., & Inciarte, A. (2022). AfroLID: A Neural Language Identification Tool for African Languages. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 1958-1981). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.128

Balazevic, I., Braun, M., & Müller, K.-R. (2016). *Language Detection For Short Text Messages In Social Media* (arXiv:1608.08515). arXiv. https://doi.org/10.48550/arXiv.1608.08515

Bañón, M., Ramírez-Sánchez, G., Zaragoza-Bernabeu, J., & Ortiz Rojas, S. (2024). FastSpell: The LangId Magic Spell. In N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, & N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* (pp. 7133-7140). ELRA and ICCL. https://aclanthology.org/2024.lrec-main.626/

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). *Enriching Word Vectors with Subword Information* (arXiv:1607.04606). arXiv. https://doi.org/10.48550/arXiv.1607.04606

Caswell, I., Breiner, T., van Esch, D., & Bapna, A. (2020). Language ID in the Wild: Unexpected Challenges on the Path to a Thousand-Language Web Text Corpus. In D. Scott, N. Bel, & C. Zong (Eds.), *Proceedings of the 28th International Conference on Computational Linguistics* (pp. 6588-6608). International Committee on Computational Linguistics. https://doi.org/10.18653/v1/2020.coling-main.579

Cavnar, W. B., & Trenkle, J. (1994). *N-gram-based text categorization*. https://www.semanticscholar.org/paper/N-gram-based-text-categorization-Cavnar-Trenkle/49af572ef8f7ea89db06d5e7b66e9369c22d7607

Céspedes, L., Kozlowski, D., Pradier, C., Sainte-Marie, M. H., Shokida, N. S., Benz, P., Poitras, C., Ninkov, A. B., Ebrahimy, S., Ayeni, P., Filali, S., Li, B., & Larivière, V. (2025). Evaluating the linguistic coverage of OpenAlex: An assessment of metadata accuracy and completeness. *Journal of the Association for Information Science and Technology*, *76*(6), 884-895. https://doi.org/10.1002/asi.24979

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). *Unsupervised Cross-lingual Representation Learning at Scale* (arXiv:1911.02116). arXiv. https://doi.org/10.48550/arXiv.1911.02116

De Giusti, M. R., Nusch, C. J., Pinto, A. V., & Villarreal, G. L. (2016). *La socialización de la producción de la Universidad Nacional de La Plata a través de su repositorio institucional SEDICI*. In Simposio Internacional: La universidad motor de transformación de la sociedad: reto de las universidades de investigación (Honorable Cámara de Senadores de la provincia de Buenos Aires, 2016). Universidad Nacional de La Plata. http://hdl.handle.net/10915/54986

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding* (arXiv:1810.04805). arXiv. https://doi.org/10.48550/arXiv.1810.04805

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., Goyal, N., Birch, T., Liptchinsky, V., Edunov, S., Grave, E., Auli, M., & Joulin, A. (2020). *Beyond English-Centric Multilingual Machine Translation* (arXiv:2010.11125). arXiv. https://doi.org/10.48550/arXiv.2010.11125

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37-54. https://doi.org/10.1609/aimag.v17i3.1230

Feldman, R., & Dagan, I. (1995). Knowledge discovery in Textual Databases (KDT). In U. Fayyad (Ed.), *KDD'95: Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 112-117). AAAI Press. https://cdn.aaai.org/KDD/1995/KDD95-012.pdf

Feng, F., Yang, Y., Cer, D., Arivazhagan, N., & Wang, W. (2022). *Language-agnostic BERT Sentence Embedding* (arXiv:2007.01852). arXiv. https://doi.org/10.48550/arXiv.2007.01852

Han, L., Erofeev, G., Sorokina, I., Gladkoff, S., & Nenadic, G. (2022). Examining Large Pre-Trained Language Models for Machine Translation: What You Don't Know about It. In P. Koehn, L. Barrault, O. Bojar, F. Bougares, R. Chatterjee, M. R. Costa-jussà, C. Federmann, M. Fishel, A. Fraser, M. Freitag, Y. Graham, R. Grundkiewicz, P. Guzman, B. Haddow, M. Huck, A. Jimeno Yepes, T. Kocmi, A. Martins, M. Morishita, … M. Zampieri (Eds.), *Proceedings of the Seventh Conference on Machine Translation (WMT)* (pp. 908-919). Association for Computational Linguistics. https://aclanthology.org/2022.wmt-1.84

Jauhiainen, T., Ranasinghe, T., & Zampieri, M. (2021). Comparing Approaches to Dravidian Language Identification. In M. Zampieri, P. Nakov, N. Ljubešić, J. Tiedemann, Y. Scherrer, & T. Jauhiainen (Eds.), *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects* (pp. 120-127). Association for Computational Linguistics. https://aclanthology.org/2021.vardial-1.14/

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016a). *FastText.zip: Compressing text classification models* (arXiv:1612.03651). arXiv. https://doi.org/10.48550/arXiv.1612.03651

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016b). *Bag of Tricks for Efficient Text Classification* (arXiv:1607.01759). arXiv. https://doi.org/10.48550/arXiv.1607.01759

Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A. F., Bogoychev, N., Martins, A. F. T., & Birch, A. (2018). Marian: Fast Neural Machine Translation in C++. In F. Liu & T. Solorio (Eds.), *Proceedings of ACL 2018, System Demonstrations* (pp. 116-121). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-4020

Lui, M., & Baldwin, T. (2011). Cross-domain Feature Selection for Language Identification. In H. Wang & D. Yarowsky (Eds.), *Proceedings of 5th International Joint Conference on Natural Language Processing* (pp. 553-561). Asian Federation of Natural Language Processing. https://aclanthology.org/I11-1062

Lui, M., & Baldwin, T. (2012). langid.py: An Off-the-shelf Language Identification Tool. In M. Zhang (Ed.), *Proceedings of the ACL 2012 System Demonstrations* (pp. 25-30). Association for Computational Linguistics. https://aclanthology.org/P12-3005/

Lui, M., Lau, J. H., & Baldwin, T. (2014). Automatic Detection and Language Identification of Multilingual Documents. *Transactions of the Association for Computational Linguistics*, *2*, 27-40. https://transacl.org/ojs/index.php/tacl/article/view/86

Mannes, J. (2016). Facebook's Artificial Intelligence Research lab releases open source fastText

on GitHub. *TechCrunch*. https://techcrunch.com/2016/08/18/facebooks-artificial-intelligence--research-lab-releases-open-source-fasttext-on-github/

Mannes, J. (2017). Facebook's fastText library is now optimized for mobile. *TechCrunch*. https://techcrunch.com/2017/05/02/facebooks-fasttext-library-is-now-optimized-for-mobile/

McNamee, P. (2005). Language identification: A solved problem suitable for undergraduate instruction. *Journal of Computing Sciences in Colleges*, *20*(3), 94-101. https://dl.acm.org/doi/10.5555/1040196.1040208

Nusch, C. J., Cagnina, L. C., Errecalde, M. L., Antonelli, L., & De Giusti, M. R. (2025). Detección de idiomas como tarea de curaduría de datos para repositorios institucionales: Desempeño de bibliotecas disponibles y modelos de lenguaje. In Garro M. (Ed.), *Actas de la Conferencia Internacional BIREDIAL-ISTEC 2024* (pp. 16-31). Universidad de Costa Rica. https://sedici.unlp.edu.ar/bitstream/handle/10915/179030/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y

Ooms, J. & Google Inc. (2023). *cld3: Google's Compact Language Detector 3* (Versión 1.6.0) [Software de computador]. https://cran.r-project.org/web/packages/cld3/

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks* (arXiv:1908.10084). arXiv. https://doi.org/10.48550/arXiv.1908.10084

Sainte-Marie, M. H., Kozlowski, D., Céspedes, L., & Larivière, V. (2025). *Sorting the Babble in Babel: Assessing the Performance of Language Detection Algorithms on the OpenAlex Database* (arXiv:2502.03627). arXiv. https://doi.org/10.48550/arXiv.2502.03627

Shuyo, N. (2010). *Language detection library for java*. Shuyo's Weblog. https://shuyo.wordpress.com/2010/11/29/language-detection-library/

Sproat, R. (1996). Multilingual text analysis for text-to-speech synthesis. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP '96* (pp. 1365-1368). IEEE. https://doi.org/10.1109/ICSLP.1996.607867

Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. En N. Calzolari, K. Choukri, T. Declerck, M. U. Doğan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)* (pp. 2214-2218). European Language Resources Association (ELRA). http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf

ANEXO 1

RESUMEN BIOGRÁFICO DE LOS AUTORES

Carlos Javier Nusch

Profesor y Licenciado en Letras por la Universidad Nacional de La Plata y Máster en Humanidades Digitales por la Universidad de Educación a Distancia de España. Ha publicado varios artículos sobre trabajo académico colaborativo, repositorios digitales, digitalización de patrimonio cultural, análisis del discurso político y literatura clásica, medieval y moderna. Trabaja en el Servicio de Difusión de la Creación Intelectual (SEDICI) de la UNLP, en el Proyecto de Enlace de Bibliotecas (PREBI) y en el repositorio CIC-Digital (CICPBA). Es miembro del Comité Asesor del Centro de Servicios en Gestión de Información (CESGI) y personal del Observatorio Medioambiental La Plata (UNLP - CICPBA - CONICET). Coordina la Oficina de Relaciones Institucionales del Consorcio Iberoamericano para la Educación en Ciencia y Tecnología (ISTEC). Participa como docente colaborador ad honorem en el curso de posgrado "Bibliotecas y Repositorios Digitales. Tecnología y aplicaciones" de la Facultad de Informática de la UNLP. Ha participado en proyectos sobre Oralidad, Escritura, Humanidades Digitales Recursos Académicos, Harvesting, OAI-PMH, Visibilidad Web, Repositorios Abiertos, Producción Académica y Científica, Accesibilidad financiados por la UNLP, la CICPBA y el ISTEC. ORCID: https://orcid.org/0000-0003-1715-4228.

Leticia Cecilia Cagnina

Doctora en Ciencias de la Computación, Magíster en Ciencias de la Computación y Licenciada en Ciencias de la Computación. Se desempeña como docente investigadora en la Universidad Nacional de San Luis (UNSL). Es Profesora Adjunta en el Departamento de Informática de la Facultad de Ciencias Físico-Matemáticas y Naturales de la UNSL. Además, es Investigadora Categoría Adjunto en la Carrera de Investigador Científico y Tecnológico del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET). Su experiencia profesional se enfoca en el campo de la Informática e Inteligencia Artificial, con especialidad en Procesamiento del Lenguaje Natural (PLN). Ha realizado importantes avances en el desarrollo y aplicación de técnicas de PLN en la bioinformática y la detección automática de riesgo en la Web. Su trayectoria académica incluye la dirección y participación en proyectos de investigación en instituciones nacionales e internacionales. Es co-directora del proyecto "Aprendizaje automático y toma de decisiones en sistemas inteligentes para la Web" y ha sido parte del proyecto "Web Information Quality Evaluation Initiative" financiado por la Unión Europea. Además, ha contribuido a proyectos relacionados con la detección de depredadores sexuales en conversaciones de chat y la evaluación de la calidad de contenido web. ORCID: https://orcid.org/0000-0001-7825-2927.

Ariel Lira

Licenciado en Informático por la Universidad Nacional de La Plata, desde 2006 forma parte del equipo de PREBI-SEDICI. Se especializa en repositorios digitales de publicaciones y datos, ciencia abierta y preservación digital. Participa en el desarrollo de herramientas y servicios para la gestión y difusión de la producción académica. Su trabajo contribuye a fortalecer el ecosistema de comunicación científica en acceso abierto. ORCID: https://orcid.org/0000-0003-3647-3101.

Gonzalo Luján Villarreal

Doctor en Ciencias Informáticas, es director de PREBI-SEDICI de la Universidad Nacional de La Plata, director del Centro de Servicios en Gestión de Información (CESGI) de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, coordinador informático de revistas científicas de la Universidad Nacional de La Plata y profesor de la Facultad de Informática de la misma universidad. ORCID: https://orcid.org/0000-0002-3602-8211.

Marcelo Luis Errecalde

Profesor Exclusivo en la Universidad Nacional de San Luis, (Argentina) y dirige el Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) de la Facultad de Cs. Físico, Matemáticas y Naturales. Trabaja desde hace más de 20 años en temáticas vinculadas a la Inteligencia Artificial, el aprendizaje automático, la minería de textos y la Web y el Procesamiento del Lenguaje Natural. Colabora con diferentes grupos líderes de España, México, Alemania, Austria y Grecia en áreas como la calidad de la información en la web, detección de plagio, detección de depredadores sexuales en la web y determinación del perfil del autor (DPA). Actualmente, el foco de atención en la DPA se centra en la determinación del género, la edad, la orientación política y los rasgos de personalidad de los autores de documentos en la Web. Como resultado de estos trabajos de investigación se han desarrollado sistemas que son actualmente los más efectivos a nivel mundial para la detección de fallas de calidad en Wikipedia y la detección anticipada de casos de depresión y anorexia en la Web. En la actualidad, sus direcciones de tesis de postgrado se centran en la detección anticipada de riesgos en la Web (depresión, suicidio, anorexia, entre otros), integración de conocimiento externo en los modelos de aprendizaje automático y transparencia e interpretabilidad de los grandes modelos del lenguaje. ORCID: https://orcid.org/0000-0001-5605-8963.

Leandro Antonelli

Obtuvo el título de Licenciado en Informática en el año 1998 momento en el cual ingresó al Centro de Investigación LIFIA. En el año 2003 obtuvo el título de Magíster en Ingeniería de

Software y en el 2012 el de Doctor en Ciencias Informáticas. Todos los títulos otorgados por la Universidad Nacional de La Plata. Leandro Antonelli se ha desempeñado tanto en la academia como en la industria. En la academia ha atravesado distintas instancias de la docencia, comenzando como ayudante allá por el año 1996. Actualmente se desempeña como profesor en materias de grado y posgrado, y también es Director de la carrera en Dirección de Proyectos de Tecnología Informatica en Universidad Abierta Interamericana. También realizó investigación principalmente en ingeniería de requerimientos, con publicaciones en conferencias nacionales e internacionales, como así también en revistas. En la industria ha trabajado en reparticiones públicas como así también en ámbitos privados (para clientes nacionales e internacionales). Se ha desempeñado en distintos roles, comenzando como desarrollador en el año 1993 y actualmente se desempeña como ingeniero de software, especializándose tanto en la gestión de requerimientos como en la gestión de proyectos en general (tanto ágiles – es Scrum Master certificado-, como tradicionales). ORCID: https://orcid.org/0000-0003-1388-0337.

Marisa Raquel De Giusti

Doctora en Ciencias Informáticas, Ingeniera en Telecomunicaciones y Profesora en Letras de la Universidad Nacional de La Plata (UNLP). Es Profesora de Posgrado en la Facultad de Informática de la UNLP, Directora del Proyecto de Enlace de Bibliotecas (PREBI, 1997) y directora del Servicio de Difusión de la Creación Intelectual (SEDICI, 2002). Impulsó la creación y fue directora hasta el año 2023 del Centro de Servicios en Gestión de Información (CESGI) de la Comisión de Investigaciones Científicas (CIC), donde actualmente reviste como Investigador Emérito. Es presidenta del Consorcio Iberoamericano para Educación en Ciencia y Tecnología (ISTEC) y Directora de la Iniciativa Library linkage (LibLink) de dicho consorcio. Integra el Comité de Expertos del Sistema Nacional de Repositorios Digitales (SNRD) y el Comité Asesor en ciencia abierta y ciudadana. Cuenta con más de 400 trabajos en áreas diversas entre las que se incluyen la gestión de la información, preservación digital, rankings y visibilidad institucional. ORCID: https://orcid.org/0000-0003-2422-6322.

Santiago Tettamanti

Licenciado en Informática por la Universidad Nacional de La Plata. Trabajó en PREBI-SEDICI de la Universidad Nacional de La Plata desde fines de 2017 hasta fines de 2022. Durante ese período, formó parte de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CICBA) como personal de apoyo. Se desempeña como Ayudante Diplomado en la Facultad de Informática de la misma universidad en las cátedras de Sistemas Paralelos y Programación Concurrente. ORCID: https://orcid.org/0000-0003-3339-7940.

ANEXO 2

REQUERIMIENTOS DE EQUIPO TÉCNICO PARA LA PRESENTACIÓN DE LA PONENCIA

Para la presentación se requerirá una computadora con conexión a internet y proyector.