Detección de idiomas como tarea de curaduría de datos para repositorios institucionales: desempeño de bibliotecas disponibles y modelos de lenguaje

Carlos Javier Nusch[1], Leticia Cecilia Cagnina[2], Marcelo Luis Errecalde[3], Leandro Antonelli[4], Marisa Raquel De Giusti[5]

Palabras claves

Repositorios Institucionales, tareas de curaduría de datos, herramientas de detección de idiomas, modelos mBERT para detección de idiomas, enfoque zero-shot

Institutional Repositories, Data Curation Tasks, Language Detection Tools, mBERT Models for Language Detection, zero-shot approach

Eje temático

Inteligencia artificial (IA) aplicada a la Ciencia Abierta

Resumen

- Presentación del problema: El enorme volumen de recursos almacenados actualmente en los repositorios digitales representa una gran dificultad a la hora de supervisar y corregir errores o mejorar la calidad de los metadatos. El presente trabajo se enfoca en la corrección del metadato idioma en los registros de resúmenes del repositorio institucional SEDICI.

- Materiales y metodología: A partir de un dataset exportado del repositorio de unos 126.081 ítems se planificó una tarea de detección automática de idiomas utilizando diferentes bibliotecas existentes compatibles con el método zero-shot (langdetect, CLD3, fastText, Polyglot, langid y TextCat). Luego se compararon los resultados obtenidos con los datos de los idiomas registrados por el personal de catalogación del repositorio. Para tratar de mejorar aún más la detección de idiomas se entrenó un modelo mBERT multilenguaje y se comparó su desempeño con el conjunto más pequeño de ítems cuya clasificación por idiomas era diferente entre humanos y la biblioteca Polyglot.

- Resultados: En general, todas las bibliotecas de detección de idiomas mostraron alrededor de un 95% de coincidencia con los idiomas identificados y catalogados por los humanos. En el caso de los modelos mBERT entrenados las coincidencias obtenidas son bajas tanto para los idiomas detectados automáticamente por Polyglot como los catalogados por humanos (78,7% y 19,6% respectivamente). Se encontraron errores de catalogación atribuibles a humanos, pero también errores de las bibliotecas o de los modelos de lenguaje en la tarea de detección.

1 Universidad Nacional de La Plata, PREBI-SEDICI carlosnusch@prebi.unlp.edu.ar
2 Consejo Nacional de Investigaciones Científicas y Técnicas
3 Consejo Nacional de Investigaciones Científicas y Técnicas
4 Universidad Nacional de La Plata, LIFIA
5 Universidad Nacional de La Plata, PREBI-SEDICI

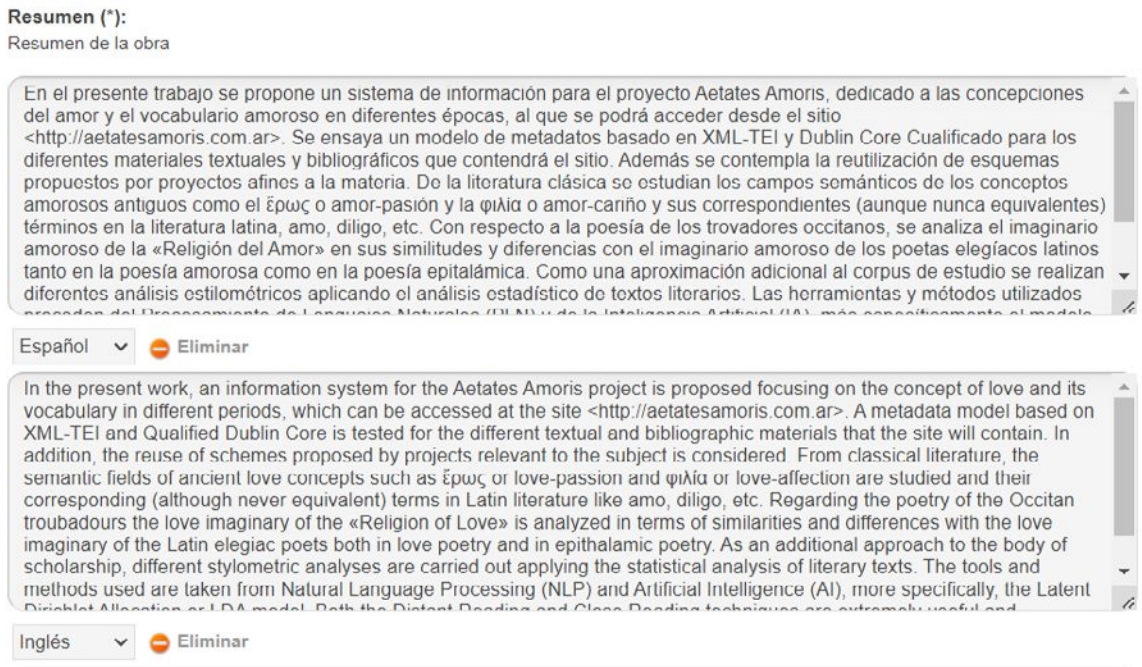
Introducción

Desde los inicios del movimiento de acceso abierto los repositorios institucionales han crecido enormemente en número y volumen de publicaciones. Tal es el caso de SEDICI, el repositorio central de la Universidad Nacional de La Plata, que ha pasado de tener 50 ítems a un año de su creación, 39.000 en 2014 y 156.299 recursos en la actualidad[6]. Entre las diferentes tareas de catalogación llevadas adelante dentro del repositorio está la de asignación del metadato idioma, tanto para el texto completo del material en cuestión como para el o los campos destinados al resumen del artículo, que puede presentarse en varios idiomas diferentes.

Dada la cantidad de campos que el personal a cargo de la catalogación de materiales debe revisar y ajustar en atención a las buenas prácticas, normas y directrices del repositorio, 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, además, se ha visto acrecentado porque el volumen de ítems que ingestan en el repositorio en tareas automáticas de importación se ha incrementado enormemente. Si bien se ha intentado simplificar y optimizar todas las tareas para llegar a reducir al mínimo estos errores, resulta imposible eliminarlos totalmente.

En las pantallas de control de datos y catalogación del software DSpace existen múltiples campos y uno de ellos es el que se destina a indicar cuál es el idioma de los resúmenes que se están registrando para cada ítem. No es algo tan extraño que se pueda pasar por alto este pequeño campo (ver Figura 1), generalmente situado debajo del campo de resumen, o que se cometa un error de cliqueo al escoger el idioma con el mouse.

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



6 Datos accesibles desde: http://sedici.unlp.edu.ar/pages/estadisticasContenidoRepositorio

Con la finalidad de explorar el grado de corrección con el que se estaba catalogando el idioma del campo resumen 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 a esa fecha. El objetivo original era llevar a cabo una tarea de curaduría automática aprovechando las diferentes herramientas de detección de idiomas disponibles en la actualidad.

El marco general de las tareas llevadas a cabo puede inscribirse 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).

Bibliotecas para la detección automática de idiomas

En las tareas de detección automática de idiomas se utilizó el lenguaje Python salvo por el caso de TextCat que se ejecutó en R. Del dataset utilizado solo se analizaron, por obvias razones, los campos de textos de los resúmenes de los diferentes ítems y las etiquetas de idioma aplicadas sobre esos campos. Se utilizaron las bibliotecas langdetect, CLD3, fastText, Polyglot, langid y TextCat con un enfoque zero-shot, esto quiere decir que no se modificaron ni re entrenaron 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 entrenamiento adicional para el conjunto de datos específico con el que se trabajó. A continuación, se detallan someramente algunas de las características de las bibliotecas de detección automática de idiomas utilizadas.

Langdetect

La biblioteca langdetect[7] es una herramienta de detección de idiomas para Python, inspirada 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. Funciona con textos de diversos dominios y 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 en la detección de idiomas.

CLD3

La biblioteca CLD3[8] (Compact Language Detector 3, sucesora de CLD1 y CLD2) es una herramienta de software desarrollada por Google que también emplea modelos de aprendizaje automático para predecir el idioma de un texto (Ooms & Google Inc, 2023). Posee soporte para más de 100 idiomas y puede procesar grandes volúmenes de texto rápidamente. Presenta una alta precisión en la detección de idiomas, inclusive con textos cortos. Puede requerir recursos computacionales mayores.

7 Disponible en: https://pypi.org/project/langdetect/
8 Disponible en: https://github.com/ropensci/cld3

Polyglot

Polyglot[9] 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 serie de tareas de PLN que exceden la mera detección (como tokenización, reconocimiento de entidades nombradas, análisis de sentimiento, traducción de palabras, etc.). Posee soporte integrado para embeddings de palabras y una serie de modelos pre entrenados lo que permite su uso inmediato sin la necesidad de entrenar modelos desde cero. Una de sus desventajas es que depende de varias bibliotecas y herramientas externas, lo que hace más ardua su instalación y configuración.

Langid

Langid[10] es una herramienta de software libre y de código abierto que puede identificar entre 97 y más de 100 idiomas diferentes (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.

TextCat

Textcat[11] es un paquete en R diseñado para la clasificación automática de textos (Hornik et al., 2013). Utiliza patrones de n-gramas para identificar la lengua en la que está escrito un texto, basándose en características estadísticas derivadas de los n-gramas que son únicos o predominantes en idiomas específicos. Se suele utilizar en tareas de procesamiento de lenguaje natural (NLP) que requieren la identificación del idioma antes de realizar análisis más profundos.

FastText

FastText[12] 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, Grave, Bojanowski, & Mikolov, 2016; Joulin, Grave, Bojanowski, Douze, et al., 2016; 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.

9 Disponible en: https://github.com/saffsd/polyglot
10 Disponible en: https://github.com/saffsd/langid.py
11 Disponible en: https://cran.r-project.org/web/packages/textcat
12 Disponible en: https://fasttext.cc/

Modelo mBERT entrenado para la detección de idiomas con el dataset de SEDICI

El modelo mBERT[13], o multilingual BERT (BERT multilingüe), es una variante del modelo BERT (Bidirectional Encoder Representations from Transformers) diseñado por Google. 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. Este modelo no se utilizó con el enfoque zero-shot ni tampoco se aplicó a la detección de idiomas de todo el dataset. Se lo entrenó con los datos detectados correctamente por la biblioteca Polyglot para examinar la posibilidad de detectar correctamente idiomas en los casos en los que las otras bibliotecas parecían no responder de la mejor manera.

Resultados preliminares

El desempeño de las diferentes bibliotecas con las que se aplicó el enfoque zero-shot fue relativamente 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 (Tabla 1). En el caso de las bibliotecas langdetect, CLD3, fastText, Polyglot y langid, se ejecutaron en un entorno de CPU provisto por Google Colab salvo para el caso de TextCat que se ejecutó localmente utilizando los recursos de una notebook. La biblioteca que mayor coincidencia tuvo en la detección de idiomas con los catalogadores humanos fue langid y la de menor tiempo de procesamiento FastText, aunque se trató de la que peores resultados obtuvo.

Tabla 1 - Porcentaje de coincidencias en la detección de idiomas y desempeño de diferentes bibliotecas

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[14]

13 Disponible en: https://github.com/google-research/bert/blob/master/multilingual.md

14 La discrepancia entre los tiempos de las otras bibliotecas y TextCat puede deberse a que fue ejecutada en una computadora local en R Studio mientras que las anteriores se corrieron en Google Colab con el lenguaje Python.

Particularidades del dataset

En las primeras pruebas de detección de idiomas con un modelo mBERT el número de predicciones correctas para los idiomas detectados por el modelo eran muy bajas. El español era confundido con el inglés y con el francés en muchos casos. El italiano no tenía predicciones correctas y tanto el francés como el alemán poseían sólo una predicción correcta cada uno. El modelo tenía serias dificultades para clasificar correctamente estas clases ya que el conjunto de datos poseía muy pocos ejemplos para el portugués, francés, alemán e italiano.

Para mejorar el rendimiento del modelo se decidió ajustar la estratificación de los datos de entrenamiento y realizar tareas de aumento de datos para las clases minoritarias. El objetivo de estas tareas era reducir el desbalance en número de ejemplos para cada clase. Además, no todos los resúmenes contaban con el metadato *idioma* (1164 no lo tenían) y por lo tanto no podía corroborarse si el idioma detectado automáticamente era o no correcto. Curiosamente, la ausencia del metadato idioma se dio en muchos de los casos en los que el lenguaje del resumen no era ninguno de los más comunes en el repositorio (español, inglés, portugués, francés, italiano o alemán).

Resultados posteriores al aumento de datos con Marian MT Model

El aumento de datos es una técnica utilizada para generar datos adicionales a partir de datos existentes. En las tareas de PLN se suele partir de textos del dataset y mediante transformaciones que generalmente buscan mantener el mismo significado del texto original, como el uso de sinónimos, por ejemplo, se generan nuevos textos. Al aumentar el conjunto de datos, se puede reducir el sobreajuste y mejorar la capacidad que presenta un modelo a la hora de generalizar con nuevos conjuntos de datos. Otro de los recursos que se suele utilizar es la traducción de textos a otros idiomas. En la tarea de aumento de datos se utilizó MarianMTModel[15] 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).

MarianMTModel 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.

15 Disponible en: https://huggingface.co/docs/transformers/model_doc/marian

Tabla 2 – Comparación de la distribución de idiomas del dataset original y las nuevas distribuciones generadas con Marian MT Model

Distribución Original			**Distribución luego del Aumento de Datos**		
Idioma	**Ejemplos**	**Porcentaje**	**Idioma**	**Ejemplos**	**Porcentaje**
es	102792	70.41	es	102789	62.69
en	39387	26.98	en	39384	24.02
pt	3346	2.29	pt	6325	3.86
fr	327	0.22	fr	6084	3.71
it	83	0.06	it	6052	3.69
de	52	0.04	de	3343	2.04

Se realizó una tarea de traducción con el modelo Marian MT incrementando las clases minoritarias a un porcentaje de alrededor del 3%. Lamentablemente, para el caso del portugués no se consiguió un modelo de traducción desde el español o el inglés que fuera compatible con la biblioteca.

Resultados luego del primer aumento de datos

Luego de obtener un mayor número de ejemplos de los idiomas de las clases minoritarias se procedió a entrenar un mBERT para clasificación de lenguajes. Con la idea de evitar el sesgo debido al desbalance de clases se redujo el número de ejemplos al número de la clase minoritaria, que luego del aumento de datos resultó ser el portugués. Se creó entonces una nueva muestra con un número igual de ejemplos para cada clase (español, inglés, francés e italiano). Luego se dividió el conjunto de datos balanceado en conjuntos de entrenamiento (12.034 ejemplos) con un porcentaje para pruebas de entrenamiento y validación. Las divisiones realizadas fueron estratificadas según la columna *idioma* manteniendo la misma proporción de clases en cada subconjunto que en el conjunto original.

Se utilizó *BertTokenizer* y *BertForSequenceClassification* para manejar la tokenización y clasificación de textos en múltiples idiomas. Se obtuvieron matrices de confusión para los conjuntos de validación y testeo. También se graficaron las Curvas de Pérdida (Loss) de entrenamiento y validación para evaluar el progreso y el rendimiento del modelo a lo largo de las diferentes épocas.

El número de épocas para el entrenamiento fue de 3 (una época completa significa que cada muestra en el conjunto de datos ha sido presentada una vez al modelo para realizar el aprendizaje). El tamaño del lote (instantaneous batch size per device), es decir, el número de muestras de datos sobre las cuales el modelo calcula la pérdida y actualiza los parámetros en una sola iteración fue de 8.

Resultados del entrenamiento del modelo mBERT

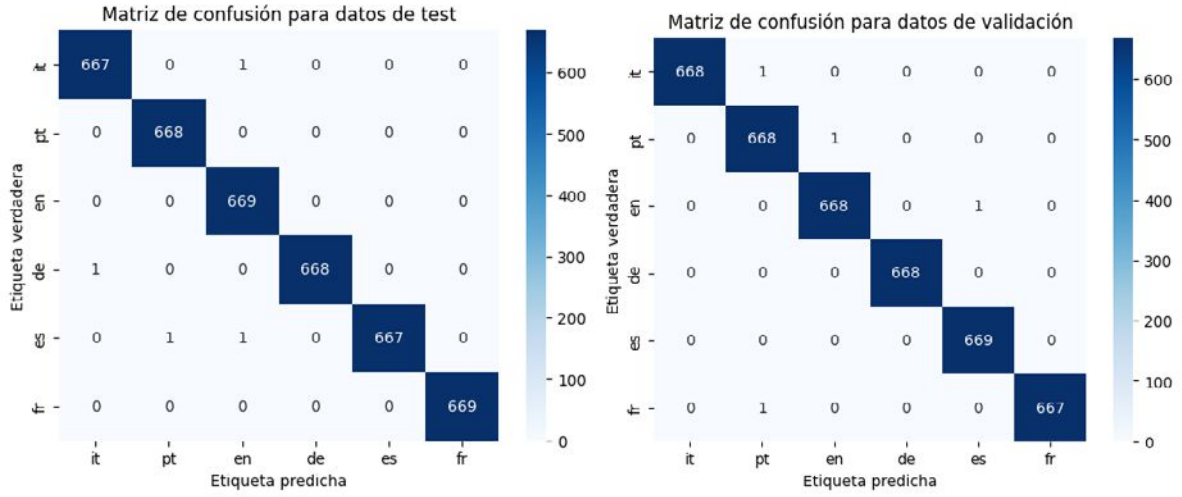
Se utilizaron varias métricas para evaluar el desempeño del modelo que se detallan a continuación:

- *Precision*: para distinguir el número de ítems correctamente identificados como pertenecientes a una clase o proporción de verdaderos positivos entre todos los ítems etiquetados como pertenecientes a esa clase.

- *Recall*: como métrica de sensibilidad del modelo para encontrar todas las instancias pertenecientes a una clase. Es la proporción de verdaderos positivos entre la suma de verdaderos positivos y falsos negativos.
- *F1-Score:* se utiliza como medida de precisión de un test y representa la media armónica de la precisión y el recall. Su valor es de 1 para precisión y recall perfectos y 0 para el peor de los desempeños.
- *Support*: es el número de ocurrencias reales de la clase en el conjunto de datos especificado.
- *Accuracy*: mide la proporción de predicciones correctas (tanto verdaderos positivos como verdaderos negativos) entre el total de casos examinados.

Tanto para los datos de prueba como para los de validación del modelo se obtuvieron precisiones muy altas en todas las clases (el modelo es muy bueno evitando falsos positivos) y los recalls fueron también altos (el modelo es efectivo en identificar todos los verdaderos positivos). El *F1-score* cercano a 1 para todas las clases indicó un buen equilibrio entre precisión y recall. La precisión general (*Accuracy*) fue de 0.999 (casi todas las predicciones del modelo fueron correctas). La consistencia entre los datos de prueba y de validación probó que el modelo generalizaba bien y no mostraba signos de sobreajuste o subajuste significativos[16].

Figura 1 - Matrices de confusión generadas luego del entrenamiento del modelo mBERT con datos aumentados al 3% para las clases minoritarias



16 En aprendizaje automático, el sobreajuste ocurre cuando un modelo aprende a identificar los datos de entrenamiento con demasiada precisión, capturando ruido o detalles irrelevantes. Esto perjudica su capacidad de generalizar a nuevos datos. El subajuste ocurre cuando un modelo es demasiado simple y no puede aprender suficientemente de la estructura subyacente de los datos de entrenamiento como para realizar buenas generalizaciones con nuevos datos.

Tabla 3 - Métricas de evaluación del modelo mBERT para los datos de validación

Reporte de clasificación				
Datos de validación				
Idioma	Precision	Recall	F1-score	Support
it	1	0.999	0.999	669
pt	0.997	0.999	0.999	669
en	0.999	0.999	0.999	669
de	1	1	0.999	668
es	0.999	1	0.999	669
fr	1	0.999	0.999	668
Accuracy			0.999	4012

Tabla 4 - Métricas de evaluación del modelo mBERT para los datos de testeo

Reporte de clasificación				
Datos de testeo				
Idioma	Precisión	Recall	F1-score	Support
it	0.999	0.999	0.999	668
pt	0.999	1	0.999	668
en	0.997	1	0.999	669
de	1	0.999	0.999	669
es	1	0.997	0.999	669
fr	1	1	1	669
Accuracy			0.999	4012

Como métrica adicional del desempeño del modelo se calculó la Pérdida de Entrenamiento (*Training Loss*) una medida que permite evaluar qué tan bien el modelo se ajusta a los datos de entrenamiento (un número más bajo indica un mejor ajuste) y la Pérdida de Validación (*Validation Loss*), una medida de qué tan bien el modelo se generaliza a nuevos datos del conjunto de validación. Durante las tres épocas de entrenamiento del modelo, la Pérdida de Entrenamiento fue consistentemente baja, lo que indica un buen ajuste a los datos de entrenamiento. Entre la primera y la segunda época, se obtuvo una notable mejora en la Pérdida de Validación (de 0.0145 a 0.0086), señal de que el modelo estaba mejorando su capacidad

de generalización. En la tercera época, la Pérdida de Validación continuó disminuyendo ligeramente (de 0.008625 a 0.008507), lo que sugiere una buena generalización sin evidencia de sobreajuste. La Pérdida de Entrenamiento alcanzó un valor extremadamente bajo (0.0001) en esta última época, lo que indica que el modelo ha aprendido casi perfectamente los datos de entrenamiento. La ligera disminución en la Pérdida de Validación entre la segunda y tercera época podría indicar que el modelo está cerca de alcanzar su mejor capacidad de generalización.

Tabla 5 - Pérdida de entrenamiento y validación a través de seis épocas durante el entrenamiento de un modelo de aprendizaje automático

Epoch	Training Loss	Validation Loss
1	0.0024	0.014507
2	0.0086	0.008625
3	0.0001	0.008507

Figura 2 - Curva Loss durante el entrenamiento y la validación

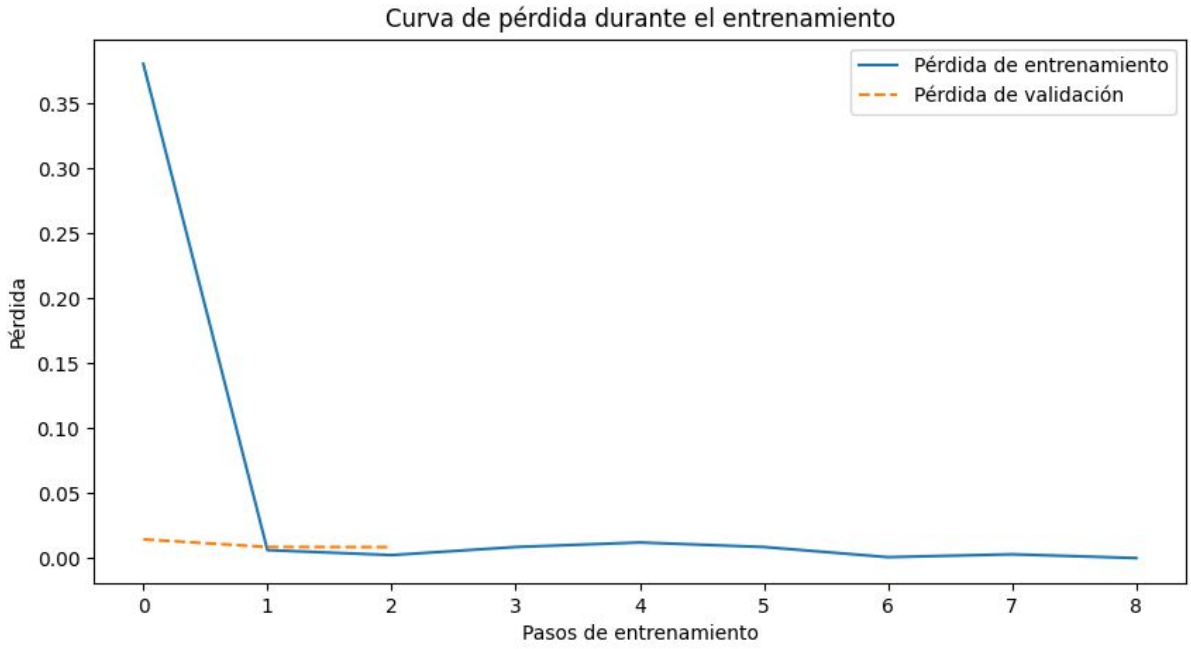


Figura 3 - Gráficos de torta con el porcentaje de coincidencia de los idiomas detectados con cada biblioteca comparado con los idiomas catalogados por humanos

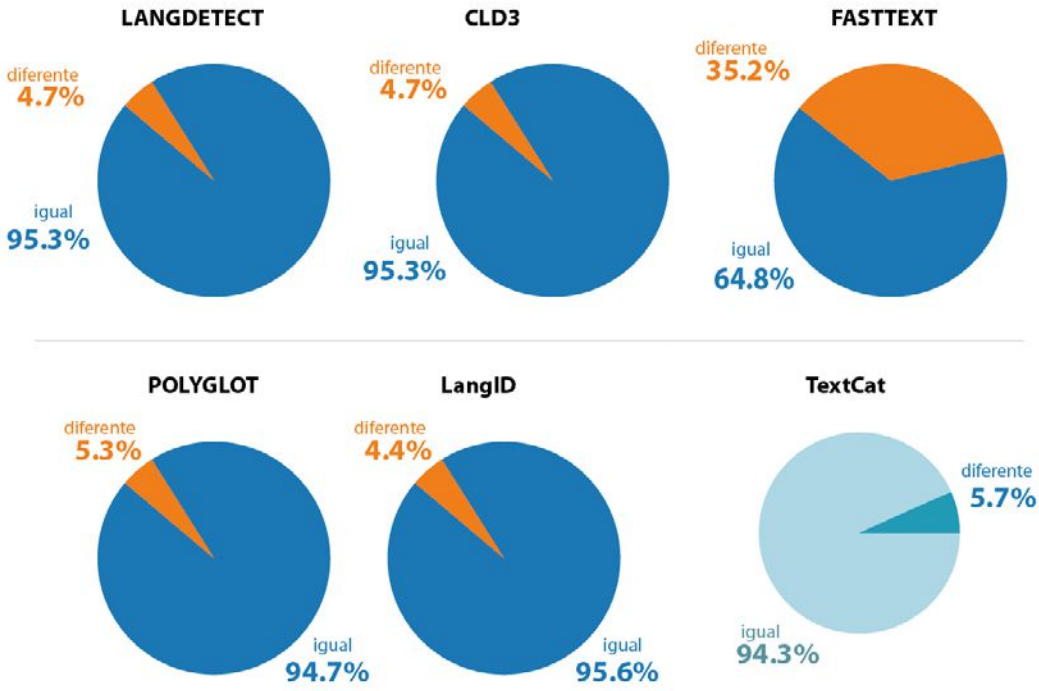
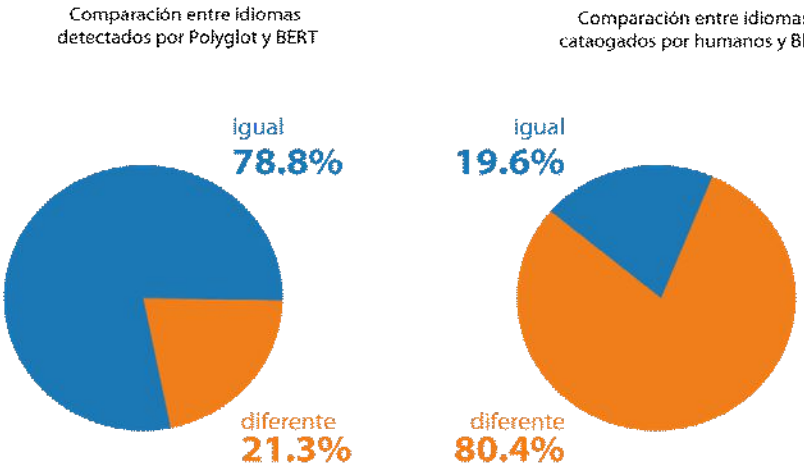


Figura 4 - Gráficos de torta con el porcentaje de coincidencia de los idiomas por mBERT comparado con los idiomas detectados por Polyglot y con los idiomas catalogados por humanos



Conclusiones

En este trabajo se presentaron diferentes resultados de tareas de detección de idiomas utilizando diferentes bibliotecas disponibles para Python y R. En su mayor parte las bibliotecas utilizadas dieron un porcentaje de coincidencia alto (alrededor del 95%) salvo por el caso de FastText. Es muy probable que el trabajo con esta biblioteca requiera entrenar modelos específicos para el conjunto de datos utilizado y también mejorar los parámetros e hiperparámetros de entrenamiento. Lo mismo ocurre con la tarea de detección de idioma que se desarrolló utilizando el modelo entrenado mBERT. Si bien el modelo mostró un excelente desempeño con los datos de entrenamiento y validación, su comparación con los datos en los cuales la detección de la biblioteca Polyglot no coincidía con lo catalogado con humanos arrojó resultados mucho menores. Esto no quiere decir, sin embargo, que el modelo funcione mal, sino que no ha sido entrenado con todos los idiomas presentes en el dataset. Una mejora en el aumento de datos o inclusive la utilización de resúmenes obtenidos de otros repositorios en diferentes lenguajes pueda mejorar el desempeño del modelo.

Otras razones también pueden explicar las fallas constantes de las diferentes bibliotecas y modelos en la detección:

1. En el conjunto de datos utilizado muchos de los resúmenes catalogados por humanos no tenían la etiqueta idioma (por motivos que se ignoran, quizá alguna falla en la migración de versiones de DSpace). Este pequeño porcentaje de idiomas como el latín, el sueco, el holandés, etc. no se encuentran representados explícitamente en las etiquetas con las que se entrenó el modelo mBERT y por lo tanto hubiera sido imposible detectarlos.
2. Algunos textos de los resúmenes simplemente tienen datos insuficientes, es decir, son pocas palabras que no alcanzan para constituir una muestra mínima para las diferentes bibliotecas y modelos.
3. En algunos casos, y con la finalidad de mejorar la visualización de los usuarios del repositorio se optó por incluir código html o LaTex (destinado a visualizar correctamente fórmulas matemáticas) en los textos de los resúmenes. Estos bloques de código seguramente introducen ruido en la detección y dificultan la tarea. Deberán ser eliminados en futuras tareas de detección para mejorar el desempeño de los modelos y bibliotecas.
4. Muchas de las bibliotecas han demostrado fallar en la detección, inclusive de los idiomas mayoritarios, cuando el texto del resumen está compuesto por un listado de palabras o frases.

En trabajos futuros se considerará también la posibilidad de utilizar y evaluar el desempeño de otros modelos de lenguaje como XLM-RoBERTa (XLM-R), Sentence-BERT (SBERT), DistilBERT o ERNIE. Una tarea importante que resta realizar pero que requerirá la intervención de etiquetadores humanos es la de re etiquetar el porcentaje de resúmenes que no cuentan con el campo de idioma y definir, cuál es la opción correcta en los casos en los que las bibliotecas y modelos no coincidieron con el idioma catalogado. Para ello, será necesario desarrollar una herramienta de interacción con catalogadores (probablemente se requiera de más de un humano para controlar los datos) que permita volver a clasificar alrededor del 5% de los ejemplos que conforman el subconjunto de datos en los que la catalogación y la detección no coincidieron. Solo una vez que se tenga la etiqueta de idioma correcta en todos los resúmenes se podrá evaluar con total certeza el desempeño de las herramientas utilizadas. Tal es el caso del modelo BERT entrenado con los datos de Polyglot, que logró un impresionante 78.7 % de coincidencia para los datos en los que las bibliotecas anteriores no coincidían con humanos y un 19,6% de coincidencia con la catalogación humana

de esos mismos datos, lo cual augura un muy buen pronóstico para el uso del modelo en tareas de detección de idiomas en el repositorio. Resta saber si para este subconjunto del dataset, fueron los humanos o las bibliotecas las que reconocieron los idiomas de mejor manera. La finalización de esta tarea que acabamos de iniciar redundará en una mucho mejor calidad de datos para el repositorio.

Bibliografía

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

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

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

Feldman, R., & Dagan, I. (1995). Knowledge discovery in Textual Databases (KDT). *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 112-117.

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. En 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

Hornik, K., Mair, P., Rauch, J., Geiger, W., Buchta, C., & Feinerer, I. (2013). The textcat Package for n-Gram Based Text Categorization in R. *Journal of Statistical Software*, *52*, 1-17. https://doi.org/10.18637/jss.v052.i06

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). *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. (2016). *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++. En 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. En 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., 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, agosto 18). 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, mayo 2). Facebook's fastText library is now optimized for mobile. *TechCrunch*. https://techcrunch.com/2017/05/02/facebooks-fasttext-library-is-now-optimized-for-mobile/

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

Shuyo, N. (2010). Language detection library for java.

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

Carlos Javier Nusch es 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 es 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

Marcelo Luis Errecalde es 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 Laboratorio de Investigación e Informática Avanzada. 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 Jefe de Trabajos Prácticos en materias de grado y como profesor en materia de posgrado. 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 es 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