Detección Automática de Idiomas

Juan Baeza Ruiz-Henestrosa
Máster Universitario en Lógica, Computación e Inteligencia Artificial
juabaerui@alum.us.es

Junio 2025

Resumen

En este trabajo se exploran tres enfoques distintos para abordar la detección automática de idiomas, utilizando un corpus multilingüe construido a partir del EuroParl Parallel Corpus: un método basado en perfiles de n-gramas, FastText y un enfoque basado en BERT multilingüe. Los tres métodos se aplican a un corpus de 60000 frases en 12 idiomas. Se implementan los algoritmos, se analizan los resultados obtenidos y se discuten las fortalezas y debilidades de cada enfoque. Todo el código está disponible públicamente en el siguiente repositorio: https://github.com/jbaezarh/language-detector.

1. Introducción y motivación

La **detección automática de idioma** (*Language Identification*, LID) es la tarea de asignar la etiqueta de idioma más probable a un fragmento de texto sin conocimiento previo del mismo. Esta tarea es esencial para aplicaciones multilingües en el procesamiento de lenguaje natural, como la traducción automática, la clasificación de textos o los sistemas de búsqueda.

En este trabajo se explorarán tres métodos distintos para abordar la tarea, utilizando un corpus multilingüe construido a partir del EuroParl Parallel Corpus [5]. Los métodos considerados representan enfoques variados y ampliamente utilizados en la literatura. El primero está basado en n-gramas, siguiendo la metodología propuesta por Cavnar y Trenkle [1], quienes aplicaron esta técnica para la categorización de textos. El segundo método se inspira en el modelo FastText [4], el cual utiliza representaciones vectoriales de n-gramas de caracteres que se combinan dentro de una arquitectura neuronal sencilla pero eficiente. Finalmente, se emplea un modelo basado en BERT multilingüe [6], el cual aprovecha el poder de los transformadores preentrenados sobre grandes volúmenes de datos en múltiples lenguas.

1.1 Objetivos

Los objetivos de este proyecto son los establecidos para el trabajo de la asignatura:

- Preparar un corpus multilingüe a partir del EuroParl Parallel Corpus.
- Implementar y comparar al menos dos modelos diferentes para la tarea de detección de idiomas
- Evaluar sistemáticamente el rendimiento de los modelos
- Analizar críticamente los resultados obtenidos.
- Familiarizarse con la literatura científica relevante en este campo.

2. Revisión de la literatura

2.1 Métodos clásicos

Uno de los enfoques más influyentes es el propuesto por Cavnar y Trenkle, basado en perfiles de ***n*-gramas de caracteres** [1]. Cada idioma se representa por un perfil ordenado de los n-gramas más frecuentes (usualmente trigramas). El idioma de un texto desconocido se predice comparando su perfil con los del corpus de entrenamiento, utilizando una distancia basada en el orden.

Otros enfoques tempranos usaron modelos **estadísticos**, como Naive Bayes o cadenas de Markov, que modelan la probabilidad de secuencias de caracteres por idioma. Estos modelos funcionan bien para textos medianos o largos, pero su rendimiento decae en entradas cortas o ruidosas.

2.2 Aprendizaje automático supervisado

Con el auge del aprendizaje automático, se popularizaron métodos como **SVM**, **Random Forests** y **redes neuronales articiales**. Estos clasificadores utilizan representaciones vectoriales del texto (como TF-IDF o embeddings) y permiten una clasificación más flexible.

Un avance relevante fue la propuesta de **FastText** [4] por Joulin et al., que combina n-gramas y embeddings para clasificación eficiente de texto. Incluye un modelo preentrenado que permite identificar más de 170 idiomas con alta precisión, incluso en frases cortas.

2.3 Modelos neuronales multilingües

Modelos **transformer multilingües**, como *mBERT* [3] o *XLM-RoBERTa* [2], entrenados con datos en decenas de idiomas, han demostrado capacidades emergentes de detección de idioma como parte de su arquitectura. Aunque no están diseñados exclusivamente para LID, se pueden adaptar fácilmente mediante una capa de clasificación.

Estos modelos son especialmente robustos ante condiciones adversas, como *code-switching*, errores ortográficos o dialectos.

2.4 Retos actuales

- **Textos cortos o ruidosos**, como tweets o títulos de noticias.
- **Lenguas similares**, especialmente entre lenguas romances o eslavas.
- **Fenómenos de code-switching**, donde múltiples idiomas se mezclan en una misma oración.
- **Idiomas con pocos recursos**, que presentan desafíos tanto para métodos clásicos como modernos.

3. Metodología empleada

En esta sección se detalla la metodología empleada en cada una de las etapas del trabajo. Todo el desarrollo se ha realizado en Python, y el código fuente está disponible públicamente en el siguiente repositorio: https://github.com/jbaezarh/language-detector.

3.1 Preparación del dataset

El primer paso ha sido la creación de un conjunto de datos adecuado para llevar a cabo la tarea de detección de idioma, cumpliendo con los requisitos establecidos. Para automatizar este proceso, se ha desarrollado la función `get_dataset`, que, a partir de las URLs correspondientes a los idiomas que se desean incluir en el corpus, el número de frases por idioma y los límites mínimo y máximo de longitud de las frases, descarga los datos necesarios y los procesa para generar un conjunto homogéneo y balanceado, listo para su uso en las siguientes etapas del trabajo.

De esta forma, hemos empleado esta función para crear un corpus balanceado formado por 60000 frases en 12 idiomas (Alemán, Español, Italiano, Francés, Esloveno, Portugués, Sueco, Checo, Danés, Polaco, Griego y Finés). Cada idioma está representado por 5000 frases extraídas de EurParl Parallel Corpus con una longitud mínima de 2 palabras y una longitud máxima de 15.

Una vez creado el dataset, lo hemos dividido aleatoriamente en las particiones de entrenamiento (70 %), validación (15 %) y test (15 %). Se emplea la misma partición a lo largo de todo el trabajo.

3.2 Análisis exploratorio de datos

En esta etapa, el objetivo ha sido analizar las características del conjunto de datos con el fin de verificar su calidad y extraer información que oriente las decisiones posteriores en el desarrollo de los modelos. Hemos estudiado los siguientes aspectos:

- Distribución del número de muestras por idioma
- Distribución de la longitud de las frases por idioma (y global)
- Palabras más comunes por idioma
- Tamaño del vocabulario por idioma
- Palabras que aparecen en varios idiomas
- Conjunto mínimo de palabras por idioma que aparecen en todas las frases de ese idioma

3.3 Modelo basado en n-gramas

Este primer enfoque se basa en el método propuesto por Cavnar y Trenkle [1], ampliamente utilizado en tareas de categorización de texto. La idea principal consiste en representar cada idioma mediante un perfil característico de n-gramas y, posteriormente, comparar nuevos textos con estos perfiles utilizando una medida de distancia. El uso de n-gramas y no palabras completas permite aplicar el modelo a palabras no vistas durante el entrenamiento.

Este modelo tiene la ventaja de ser sencillo, interpretable y eficiente computacionalmente, especialmente útil como línea base para comparar con métodos más complejos.

Entrenamiento. Durante la fase de entrenamiento, se construye un perfil para cada idioma a partir del corpus previamente generado. Un perfil se define como una lista ordenada de los n-gramas más frecuentes encontrados en los textos de ese idioma. En la implementación original se consideran n-gramas de longitud 1 hasta 5, y se conservan únicamente los 300 más frecuentes, aunque nosotros hemos experimentado diversas combinaciones de estos hiperparámetros. Esta representación permite capturar patrones característicos de cada lengua sin depender de un análisis gramatical complejo.

Predicción. Para determinar el idioma de una nueva frase, primero se genera su perfil de n-gramas utilizando los mismos criterios que en el entrenamiento. A continuación, se calcula la distancia entre el perfil de la frase y los perfiles conocidos de cada idioma. Se emplea la distancia *out-of-place*, que mide la discrepancia en el orden de aparición de los n-gramas comunes entre dos perfiles. El idioma con la menor distancia es el seleccionado como predicción.

Preprocesamiento. Antes de generar los perfiles, los textos se normalizan mediante un preprocesamiento sencillo pero efectivo, en línea con el planteamiento original del modelo:

- Se convierten todos los caracteres a minúsculas.
- Se eliminan los signos de puntuación y los dígitos, preservando únicamente letras y apóstrofes.
- Los espacios no se eliminan, ya que son relevantes para la generación de ciertos n-gramas.
- Se añade padding al principio y al final de la frase, con el fin de permitir la extracción de n-gramas que capturen los bordes de la secuencia completa.

Implementación. Tanto el modelo como el preprocesamiento se han encapsulado en la clase `NGramLanguageIdentifier` que incorpora los métodos `fit` y `predict`.

3.4 Modelo probabilístico-neuronal: FastText

La siguiente implementación se basa en el trabajo de Joulin et al. titulado Bag of Tricks for Efficient Text Classification [4]. Este modelo neuronal sencillo extiende la idea de los perfiles de n-gramas al paradigma vectorial del lenguaje, permitiendo una mejor generalización y aprovechamiento de las características presentes en los datos.

El modelo trabaja sobre distribuciones de n-gramas con el objetivo de generalizar incluso a palabras no vistas durante el entrenamiento. En lugar de comparar perfiles de n-gramas directamente, se aprende un *embedding* para cada n-grama, y la representación de una frase se obtiene promediando los *embeddings* de sus n-gramas constituyentes. Esta representación se alimenta a un clasificador lineal (capa softmax) para predecir el idioma.

Preprocesamiento. El texto se procesa de la siguiente forma:

- Conversión a minúsculas.
- Eliminación de puntuación y caracteres no deseados, conservando únicamente letras Unicode, espacios y apóstrofes.
- Generación de n-gramas de tamaño 1 a 5 para cada palabra, empleando delimitadores especiales ('<','>') para marcar los extremos de las palabras.

Representación y arquitectura. El vocabulario consiste en los n-gramas más frecuentes (hasta un máximo predefinido). Cada n-grama tiene asignado un vector *embedding* aprendido durante el entrenamiento. La representación de una frase se calcula como la media de los *embeddings* de sus n-gramas. Sobre esta representación se aplica una capa densa con activación softmax para obtener la distribución de probabilidad sobre las clases (idiomas).

Entrenamiento. El modelo se entrena minimizando la entropía cruzada entre la predicción y la etiqueta verdadera mediante el optimizador Adam. Se utiliza un esquema de *early stopping* basado en la métrica de tasa de acierto sobre el conjunto de validación para evitar el sobreajuste. Aunque en el artículo original se emplea *hierarchical softmax* para acelerar el entrenamiento en casos con muchas clases, en este trabajo no se implementó debido al número reducido de idiomas considerados.

Implementación. El modelo se ha implementado en Python utilizando TensorFlow y Keras. El preprocesamiento y la generación del vocabulario de n-gramas están encapsulados dentro de la clase `FastTextClassifier`, que expone los métodos estándar `fit` y `predict` para facilitar el entrenamiento y la inferencia.

3.5 Modelo basado en transformers

Con el objetivo de superar las limitaciones detectadas en los modelos previos —principalmente la ausencia de mecanismos de atención y la dependencia exclusiva de n-gramas— implementamos un enfoque basado en transformers utilizando el modelo preentrenado `distilbert-base-multilingual-cased` [6]. Esta versión ligera de BERT, entrenada en 104 idiomas, asegura una amplia cobertura que incluye todos los presentes en nuestro conjunto de datos. Para ello, entrenamos una capa clasificadora específica y realizamos fine-tuning sobre el modelo para adaptar sus representaciones a nuestra tarea.

Preprocesamiento En esta ocasión optamos por un preprocesamiento distinto al de los métodos anteriores. Como el modelo BERT utilizado es sensible a las mayúsculas y en algunos idiomas el uso de las mismas es un rasgo característico, decidimos no convertir todo el texto a minúsculas. Además, mantuvimos los signos de puntuación para preservar información útil sobre la estructura y estilo del idioma. Por otro lado, sí eliminamos los números, ya que no aportan información relevante para la detección del idioma.

Representación y arquitectura. El modelo utiliza como base el encoder preentrenado de DistilBERT, cuyos pesos se mantienen congelados inicialmente. Para la representación de cada secuencia, se aplica una capa de *mean pooling* que calcula la media de los embeddings de todos los tokens, considerando la máscara de atención para ignorar el padding. Esta representación se alimenta a una cabeza clasificadora entrenable encargada de predecir el idioma. Posteriormente, se realiza fine-tuning del modelo descongelando las últimas 2 capas del encoder y entrenándolas conjuntamente con la cabeza para mejorar el rendimiento.

Entrenamiento. El entrenamiento se realiza en dos fases. Primero, se entrena únicamente la cabeza clasificadora,

manteniendo congelados los pesos del encoder para estabilizar el aprendizaje inicial. Posteriormente, se descongelan los dos últimos bloques del encoder para hacer un fine-tuning parcial, lo que permite adaptar mejor el modelo a la tarea sin un riesgo elevado de sobreajuste. En todo el proceso se emplea el optimizador AdamW y un tamaño de batch de 16.

3.6 Técnicas de evaluación

Evaluamos el rendimiento de los modelos sobre la partición de test utilizando métricas estándar de clasificación: accuracy, precision, recall y F1-score, tanto a nivel global como por clase. Además, analizamos el rendimiento en función de la longitud de las frases, agrupándolas en categorías (corta, media y larga) para detectar posibles diferencias según el tamaño de la entrada. También examinamos la matriz de confusión de cada modelo.

Adicionalmente, se lleva a cabo un análisis de los errores de cada modelo, para detectar las causas y conocer las limitaciones de cada enfoque.

4. Experimentación y resultados

En esta sección se comentan los experimentos realizados, las configuraciones de hiperparámetros empleadas y los resultados obtenidos. Las tablas con las métricas obtenidas en la experimentación para los distintos modelos desagregando por idioma y tamaño de la frase se encuentran en el Apéndice.

4.1 Análisis exploratorio de datos

En esta sección se trabaja solo con el conjunto de entrenamiento. Se comprueba que es un conjunto balanceado. En la Figura 1 puede verse la longitud de las frases según el idioma; se comprueba que no hay grandes diferencias entre los idiomas. En la Figura 2 podemos ver el tamaño del vocabulario de cada uno de los idiomas considerados; en nuestro corpus las lenguas con un vocabulario más amplio son el finés, el polaco, el checo y el eslovaco. En el análisis se observa también que hay palabras que aparecen en varios idiomas; podemos diferenciar nombres propios de personas, nombres de países, abreviaturas políticas e institucionales, términos comunes en la política europea, artículos o preposiciones frecuentes y términos comunes en varios idiomas. En la Figura 3 vemos el número de palabras compartidas entre cada par de idiomas; según este criterio, las lenguas que presentan mayor similitud léxica son el español y el portugués, el danés y el sueco, el portugués y el italiano, así como el español y el italiano. No hay palabras que aparezcan en todas las frases de un idioma, ni siquiera en las de mayor longitud.

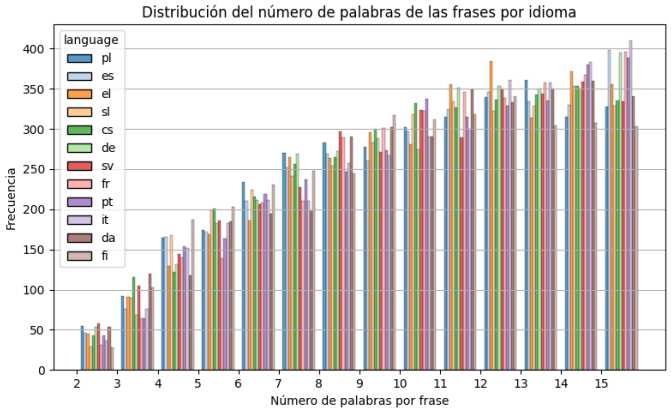

Figura 1: Distribución del número de palabras por frase según el idioma

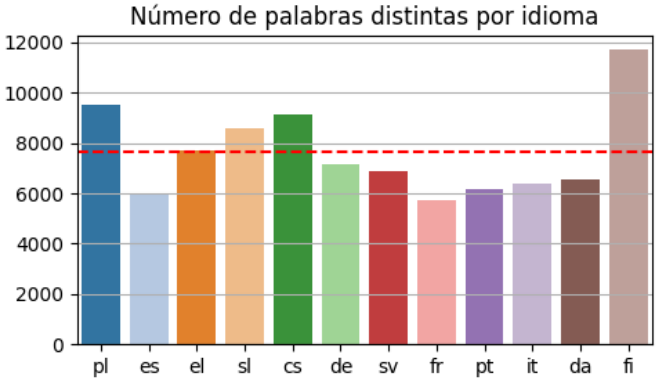

Figura 2: Tamaño del vocabulario según el idioma

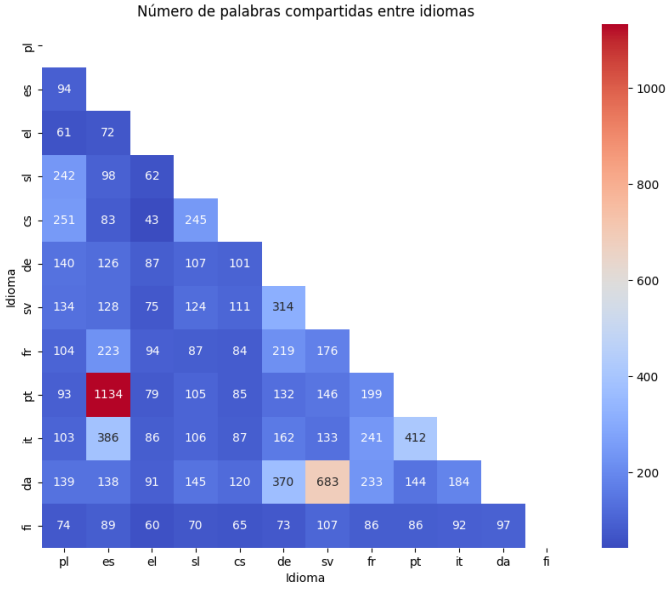

Figura 3: Número de palabras comunes entre idiomas

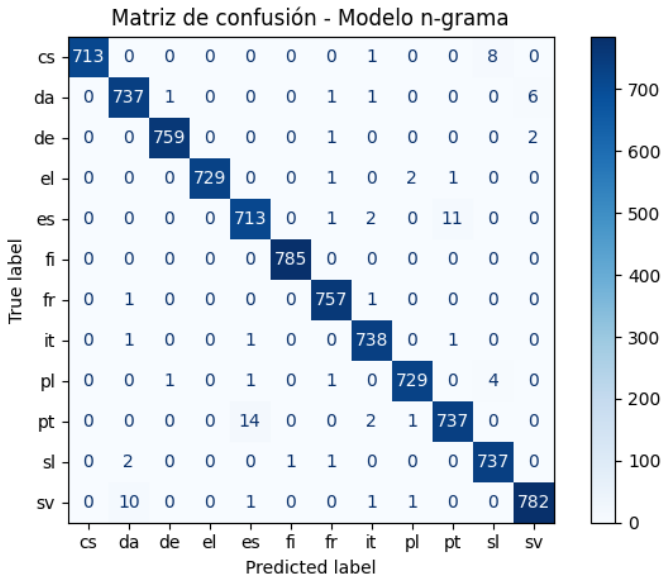

Figura 4: Matriz de confusión del modelo de n-grama sobre los datos de test.

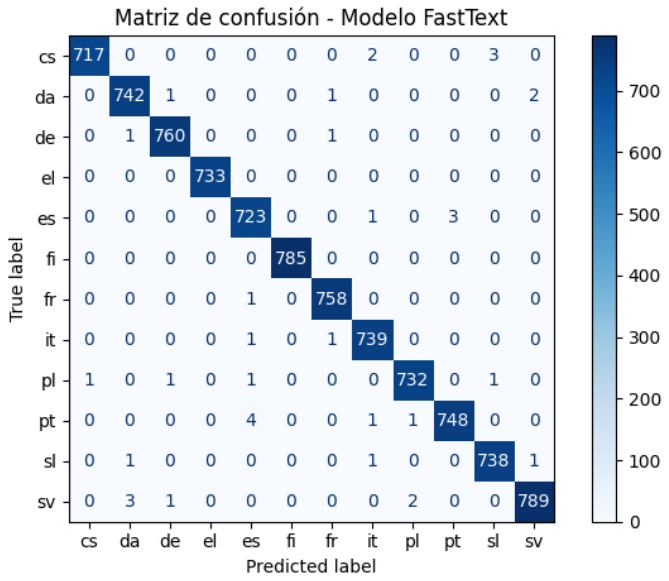

Figura 5: Matriz de confusión del modelo FastText sobre los datos de test.

4.2 Modelo basado en n-gramas

Tras probar distintas combinaciones de hiperparámetros, se selecciona el modelo que utiliza n-gramas de longitud entre 1 y 5, con un perfil de n-gramas de tamaño 1000. El entrenamiento de este modelo dura aproximadamente 3 segundos en el equipo empleado, mientras que la predicción sobre las 9001 frases del conjunto de test toma alrededor de 1808 milisegundos. La tasa de acierto del modelo sobre el conjunto test es del 0.9906 y el F1 medio es de 0.9905. En la Figura 4 puede verse la matriz de confusión del modelo sobre el conjunto de test.

Análisis de errores. Se detectan errores en el dataset por frases mal etiquetadas y textos muy cortos o ambiguos. Algunas confusiones se explican por la gran similitud entre idiomas o expresiones comunes, especialmente entre lenguas cercanas como español y portugués. En general, los errores del modelo suelen presentar puntuaciones muy similares entre la predicción y el idioma correcto. Los errores aumentan en las frases más cortas.

4.3 Modelo probabilístico-vectorial

Los resultados aquí presentados corresponden al modelo configurado con un vocabulario de tamaño 10,000 y utilizando n-gramas de longitud entre 1 y 6. Esta configuración ha demostrado ofrecer el mejor compromiso entre latencia y rendimiento tras una fase exhaustiva de experimentación.

El entrenamiento se llevó a cabo empleando *early stopping* y un tamaño de lote de 32. En el hardware utilizado, el tiempo de entrenamiento fue de aproximadamente un minuto, mientras que la inferencia sobre las 9001 frases del conjunto de prueba se completó en 1303 milisegundos.

El modelo alcanzó una tasa de acierto de 0.9959 y un valor F1 medio también de 0.9959 sobre el conjunto de test. La Figura 5 muestra la matriz de confusión correspondiente.

Análisis de errores. Los errores del modelo pueden clasificarse en tres categorías principales: errores atribuibles al conjunto de datos, errores sistemáticos y errores interpretables. Entre los primeros, se encuentran ejemplos mal etiquetados en el corpus de entrenamiento, como frases en español o inglés anotadas incorrectamente como polaco, checo o esloveno. También se identifican casos con textos breves o altamente ambiguos, que presentan formas idénticas en múltiples lenguas; por ejemplo, “É anti-social.” podría corresponder tanto al portugués como al italiano. En cuanto a los errores sistemáticos, se observa la clasificación incorrecta de frases inequívocas, como “Permítanme que les explique tres de ellas.”, identificada como francés pese a contener estructuras morfosintácticas exclusivas del español. En este tipo de errores, el modelo ignora señales claras como el uso de tildes, diéresis o signos de puntuación propios de un idioma. Por último, se encuentran errores interpretables, donde la confusión es comprensible desde una perspectiva superficial. Es el caso de “Teroristi, morilci in komunisti!”, clasificada como italiano debido a la similitud fonológica y morfológica de ciertos sufijos, aunque la frase pertenece al esloveno.

4.4 Modelo basado en *Transformers*

Se emplea el modelo `distilbert-base-multilingual-cased`. La arquitectura se completa con una cabeza clasificadora compuesta por una capa densa de 256 neuronas con activación *ReLU*, seguida de una capa de salida con

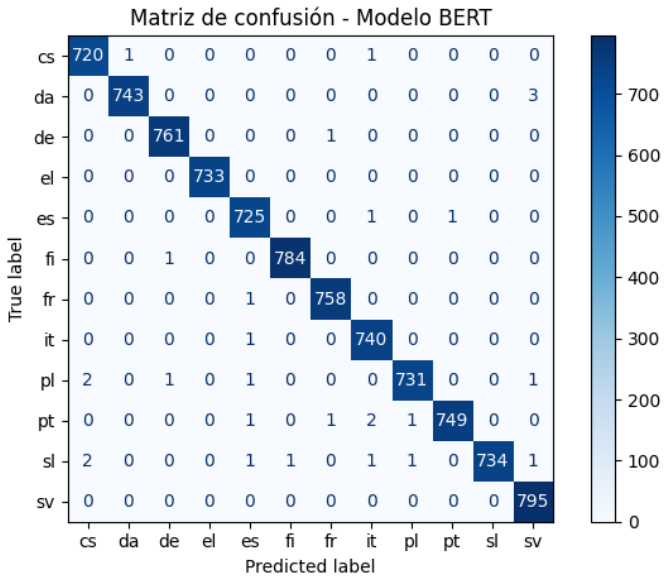

Figura 6: Matriz de confusión del modelo BERT sobre los datos de test.

activación *softmax*. Se aplica *Dropout* con una tasa de 0.2 para mitigar el sobreajuste.

Durante el entrenamiento, únicamente se ajustan los pesos de la cabeza clasificadora, empleando la estrategia de *early stopping* para evitar el sobreentrenamiento. Se emplea la función de pérdida de entropía cruzada, el optimizador AdamW con una tasa de aprendizaje de 10^{-4} y un tamaño de lote de 16. En el hardware utilizado, el tiempo de entrenamiento fue de aproximadamente 10 minutos, mientras que la inferencia sobre las 9001 frases del conjunto de prueba se completó en 50000 milisegundos.

En experimentaciones preliminares, se evaluó el *fine-tuning* de los dos últimos bloques del modelo base. Sin embargo, esta configuración provocó un sobreajuste inmediato al conjunto de entrenamiento, lo que motivó la decisión de mantener el modelo base congelado y limitar el ajuste únicamente a la cabeza clasificadora.

El modelo alcanzó una tasa de acierto de 0.9969 y un valor F1 medio también de 0.9959 sobre el conjunto de test. La Figura 6 muestra la matriz de confusión correspondiente.

Análisis de errores. Se observan varias fuentes recurrentes de confusión. Por un lado, las frases que contienen elementos multilingües tienden a generar ambigüedad, como en el caso de "Klimawandel (annonce de propositions de résolution déposées): siehe Protokoll", donde coexisten alemán y francés (el modelo detecta estos dos idiomas como los más probables). Asimismo, los nombres propios influyen en la predicción del modelo: frases como "Poročilo: Miroslav Ouzký (A6-0395-2007)" y "Poročilo: Inés Ayala Sender" están etiquetadas como esloveno, pero el modelo las clasifica como checo y español, respectivamente, en función del nombre incluido (aunque el segundo idioma con mayor score es el esloveno). También se detecta una tendencia a la confusión entre lenguas escandinavas (danés, sueco, finés) y entre lenguas romances (italiano, español, portugués), lo que sugiere solapamiento en los patrones léxicos aprendidos. Por otro lado, se identifican nuevos errores en el propio corpus, como "Relazione: Herczog" mal etiquetado como checo, pese a que "relazione" es italiano. Finalmente, ciertas frases breves o mal puntuadas producen errores arbitrarios, como "(Posiedzenie zostało otwarte o godz." clasificada como polaco en un caso y como checo en otro muy similar, lo que indica que los signos de puntuación o su ausencia pueden inducir sesgos inesperados.

5. Análisis crítico y comparativo

En este trabajo se han explorado tres enfoques diferentes para abordar el problema de la identificación del idioma en textos de distinta longitud empleando un corpus extraído del EuroParl Parallel Corpus.

El primer enfoque adoptado ha sido un enfoque basado en perfiles de n-gramas. La principal ventaja de este modelo es su simplicidad y su fácil interpretación. Esta es, sin embargo, su principal limitación, ya que no captura adecuadamente la estructura lingüística más allá de patrones superficiales de caracteres. Como resultado, su rendimiento disminuye en textos muy cortos o cuando existen similitudes léxicas entre idiomas, especialmente entre lenguas de la misma familia. Siendo consciente de esta limitación, hemos empleado un tamaño de perfil de n-gramas 3 veces mayor al empleado por [1]. Pero esto no soluciona el problema, tan solo lo amortigua, a costa de aumentar el costo computacional del modelo y su latencia. Las limitaciones mencionadas no se solucionan aumentando el tamaño del modelo, son inherentes a este por la forma en la que trabaja. Por ejemplo, aun considerando un tamaño del perfil de n-gramas de 5000 el modelo no es capaz de detectar que la palabra "cómo" es española. Pese a todo, hemos visto que hemos obtenido muy buenos resultados con este modelo.

Con el objetivo de intentar mitigar los inconvenientes de este modelo manteniendo el enfoque basado en n-gramas, probamos con el modelo FastText que emplea embeddings de los n-gramas para sustituir los perfiles de n-gramas, que ya hemos visto que tienen más problemas a la hora de trabajar con textos más cortos o ambiguos. Con este modelo no solo logramos conseguir una mejora en rendimiento sino también en velocidad. Este modelo clasifica perfectamente todas las frases de más de 12 palabras y lograba una tasa de acierto de 0.9825 en frases de menos de 7 palabras. Pese a ello, el modelo comete errores entre lenguas muy parecidas al solo considerar un vocabulario finito de n-gramas. Aumentar el tamaño del vocabulario los mitiga pero no soluciona el problema de raíz.

Con el objetivo de superar estas limitaciones, se empleó un modelo BERT multilingüe preentrenado en 104 idiomas para abordar la tarea. Este enfoque obtuvo las métricas

más altas en todos los tamaños de frase y en casi todas las lenguas analizadas. El modelo muestra una especial robustez en escenarios con texto ruidoso, frases muy similares en diferentes idiomas o presencia de múltiples lenguas en una misma oración. No obstante, presenta una desventaja significativa frente a los modelos más ligeros: su elevada latencia, que resulta ser aproximadamente 40 veces mayor que la de FastText.

En resumen, cada uno de los enfoques evaluados presenta ventajas y limitaciones que los hacen más o menos adecuados según el contexto de aplicación. Los modelos basados en perfiles de n-gramas ofrecen una solución simple y razonablemente eficaz para textos largos y entornos con recursos limitados. FastText representa un equilibrio adecuado entre rendimiento y eficiencia, siendo especialmente útil en escenarios de producción con restricciones de latencia. Finalmente, BERT destaca por su precisión y robustez frente a ambigüedades y ruido lingüístico, aunque su uso queda limitado a contextos donde la latencia y el coste computacional no sean factores críticos. Esta comparación evidencia que la elección del modelo debe guiarse no solo por su rendimiento en métricas estándar, sino también por las necesidades prácticas del entorno en el que se desea desplegar.

6. Conclusiones y propuestas de mejora

A lo largo de este trabajo hemos presentado tres enfoques para abordar el problema de la identificación del idioma. Hemos desarrollado una función que nos permite crear datasets de forma automática a partir del EuroParl Parallel Corpus en base a nuestras especificaciones. Hemos empleado esta función para crear un corpus cons 60000 frases y 12 idiomas, el cual hemos analizado en profundidad y lo hemos empleado para entrenar 3 modelos diferentes y conocer las fortalezas y limitaciones de cada uno de ellos. Hemos visto que cada modelo responde a unas necesidades prácticas bien diferenciadas, todos ellos son aplicables a la realidad.

7. Conclusiones y propuestas de mejora

En este trabajo se han explorado tres enfoques distintos para abordar el problema de la identificación automática del idioma en textos de diferente longitud. Para ello, se desarrolló una función que permite generar de forma automatizada conjuntos de datos a partir del EuroParl Parallel Corpus, adaptándolos a las especificaciones deseadas. Utilizando esta herramienta, se construyó un corpus de 60000 frases en 12 idiomas, el cual fue analizado y empleado para entrenar y evaluar tres modelos representativos: uno basado en perfiles de n-gramas, otro con FastText y un tercero basado en BERT multilingüe.

Los resultados obtenidos muestran que cada modelo responde a diferentes necesidades prácticas: el modelo de n-gramas destaca por su simplicidad y eficiencia en contextos con recursos limitados; FastText ofrece un equilibrio notable entre rendimiento y velocidad, adecuado para aplicaciones en tiempo real; mientras que BERT proporciona el mayor grado de precisión, especialmente en contextos multilingües, ambiguos o ruidosos, aunque a costa de una latencia significativamente mayor. Todos ellos han demostrado ser viables, dependiendo del contexto y las restricciones del caso de uso.

Como líneas futuras de trabajo, proponemos escalar los experimentos a la totalidad de idiomas disponibles en el EuroParl Parallel Corpus, lo cual requerirá una mayor capacidad de cómputo. No obstante, se ha observado que ciertos errores en la clasificación provienen del ruido presente en el corpus, por lo que una curación previa de los datos sería recomendable. Esta curación podría apoyarse en los propios modelos entrenados, empleando sus predicciones para detectar frases potencialmente mal etiquetadas.

Además, se ha identificado que los nombres propios de persona introducen sesgos en el proceso de clasificación. Sería útil implementar un sistema automático de detección y anonimización de estos elementos durante el preprocesamiento, con el objetivo de entrenar modelos más robustos y generalizables.

Referencias

[1] Cavnar, W. B., and Trenkle, J. M. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (1994), pp. 161–175.

[2] Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *CoRR abs/1911.02116* (2019).

[3] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR abs/1810.04805* (2018).

[4] Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759* (2017).

[5] Koehn, P. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit* (Phuket, Thailand, 2005), AAMT, AAMT, pp. 79–86.

[6] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv abs/1910.01108* (2019).

Apéndice

Aquí se recogen las métricas obtenidas por los modelos sobre el conjunto de test.

Cuadro 1: Métricas de rendimiento globales por modelo.

	accuracy	precision_macro	recall_macro	f1_macro
n-grams	0.9906	0.9906	0.9905	0.9905
fasttext	0.9959	0.9959	0.9959	0.9959
bert	0.9969	0.9969	0.9969	0.9969

Cuadro 2: Métricas de rendimiento por idioma según el modelo.

	n-grams			fasttext			bert		
	precision	recall	f1	precision	recall	f1	precision	recall	f1
label									
cs	1.0000	0.9875	0.9937	0.9986	0.9931	0.9958	0.9945	0.9972	0.9959
da	0.9814	0.9879	0.9846	0.9933	0.9946	0.9940	0.9987	0.9960	0.9973
de	0.9974	0.9961	0.9967	0.9961	0.9974	0.9967	0.9974	0.9987	0.9980
el	1.0000	0.9945	0.9973	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
es	0.9767	0.9807	0.9787	0.9904	0.9945	0.9925	0.9932	0.9972	0.9952
fi	0.9987	1.0000	0.9994	1.0000	1.0000	1.0000	0.9987	0.9987	0.9987
fr	0.9921	0.9974	0.9947	0.9961	0.9987	0.9974	0.9974	0.9987	0.9980
it	0.9893	0.9960	0.9926	0.9933	0.9973	0.9953	0.9933	0.9987	0.9960
pl	0.9945	0.9905	0.9925	0.9959	0.9946	0.9952	0.9973	0.9932	0.9952
pt	0.9827	0.9775	0.9801	0.9960	0.9920	0.9940	0.9987	0.9934	0.9960
sl	0.9840	0.9946	0.9893	0.9946	0.9960	0.9953	1.0000	0.9906	0.9953
sv	0.9899	0.9836	0.9868	0.9962	0.9925	0.9943	0.9938	1.0000	0.9969

Cuadro 3: Métricas de rendimiento globales por tamaño según el modelo.

	n-grams		fasttext		bert	
	accuracy	f1_macro	accuracy	f1_macro	accuracy	f1_macro
size						
large	0.9996	0.9996	1.0000	1.0000	1.0000	1.0000
medium	0.9969	0.9969	0.9987	0.9986	0.9993	0.9993
small	0.9606	0.9597	0.9825	0.9821	0.9859	0.9856