Vol. 1, Año 4, No. 12, 2025
ISSN 2954-4211

Revista Internacional Socio-Innova-Tec del Altiplano

From 'Computadora' to 'Ordenador': The Impact of Spanish Dialects on LLM Classification Variance

De 'Computadora' a 'Ordenador': El Impacto de los Dialectos del Español en la Varianza de Clasificación de los LLM

Kevin Gyovani RAMÍREZ-VITE[1], José Daniel RUPERTO-VILLALPANDO[1*], Dulce Arisbeth CÓRDOBA-BELTRÁN[1], Elizabeth VÁZQUEZ-MUNIVE[1]

[1]*TecNM- Tecnológico de Estudios Superiores de Chalco, Ingeniería Informática, Chalco, Estado de México, México.*
(0009-0006-5886-2123**,** 0009-0002-7691-8435**,** 0009-0000-7274-4875, 0009-0009-7944-3958**)**

Sent date: 31/October/2025 Acceptance date: 27/November/2025

Abstract:

This study investigates the impact of Spanish dialectal variation on the performance and reproducibility of Large Language Models (LLMs) in text classification tasks. Specifically, it compares Peninsular Spanish and Mexican Spanish using benchmark datasets for sentiment analysis and fake news detection. The methodology follows the CRISP-DM framework, emphasizing data normalization, dialectal adaptation, and controlled model evaluation through multiple training runs. Models such as RoBERTa were fine-tuned and tested to quantify both intra-model and inter-dialectal variance. Results show that models trained on Peninsular Spanish achieved higher stability and accuracy in sentiment analysis, while those trained on Mexican Spanish performed better in fake news detection. These differences reveal that dialectal variation significantly influences model behavior and highlight the limitations of relying solely on one Spanish variant for NLP tasks. The findings underscore the importance of developing balanced and representative datasets that capture the linguistic diversity of Spanish, ultimately contributing to fairer and more reliable LLMs for multilingual applications.

Keywords**:** LLMs, Dialectal Variation, Classification, NPL, Corpus.

Resumen:

Este estudio investiga el impacto de la variación dialectal del español en el desempeño y la reproducibilidad de los Modelos de Lenguaje de Gran Escala (LLMs) en tareas de clasificación de texto. En particular, se comparan el español peninsular y el español de México utilizando conjuntos de datos de referencia para análisis de sentimientos y detección de noticias falsas. La metodología sigue el marco CRISP-DM, con énfasis en la normalización de datos, la adaptación dialectal y la

evaluación controlada de los modelos mediante múltiples ejecuciones de entrenamiento. Se ajustaron y probaron modelos como RoBERTa para cuantificar la varianza intra-modelo e interdialectal. Los resultados muestran que los modelos entrenados con español peninsular alcanzaron mayor estabilidad y precisión en análisis de sentimientos, mientras que los entrenados con español mexicano obtuvieron mejores resultados en detección de noticias falsas. Estas diferencias evidencian que la variación dialectal influye significativamente en el comportamiento de los modelos y subrayan las limitaciones de basarse en una sola variante del español para tareas de PLN. Los hallazgos destacan la importancia de desarrollar conjuntos de datos equilibrados y representativos que reflejen la diversidad lingüística del español, contribuyendo así a modelos más justos y confiables.

***Palabras clave*:** LLMs, Variación dialectal, Clasificación, PLN, Corpus.

Corresponding author E-mail: jose_rv1@tesch.edu.mx
Tel: +52 5540975304

1. Introducción

En los últimos años, los modelos de lenguaje de gran escala (LLMs, por sus siglas en inglés) se han convertido en una herramienta clave dentro del procesamiento de lenguaje natural (NLP). Gracias a ellos, hoy es posible realizar con gran precisión tareas como la clasificación automática de textos, el análisis de sentimientos o la detección de noticias falsas. Estos avances han transformado la forma en que se interactua con la información digital y han abierto nuevas posibilidades tanto en la investigación como en aplicaciones prácticas. A pesar de estos hallazgos, existe un vacío de conocimiento respecto al comportamiento de los modelos de lenguaje de gran escala LLMs, en lenguas con alta diversidad dialectal, como el español. Diferencias léxicas y semánticas entre variantes, por ejemplo, “ordenador” o “computadora” o “coche” o “carro” podrían afectar la performance de los modelos y su confiabilidad en aplicaciones prácticas.

Este estudio busca cuantificar cómo las variantes dialectales del español impactan la clasificación automática de texto. Se comparan dos variantes principales: el español de España y el español de México, utilizando modelos de referencia y LLMs de propósito general. La pregunta central de investigación es:

¿Cuál es el impacto de las variantes dialectales del español (Peninsular vs. Mexicano) en la performance y la reproducibilidad de los modelos de lenguaje de gran escala en tareas de clasificación de texto?

Con dicha investigación no solo se busca evidenciar un posible sesgo lingüístico en el procesamiento automático del español, sino también aportar a la construcción de modelos y datasets más representativos y justos para toda la comunidad hispanohablante.

2. Estado del arte

La variabilidad en el desempeño de los LLMs es una preocupación creciente en la investigación de procesamiento de lenguaje natural (NLP). Como menciona YanXue1, et al.,

en la obra titulada We Need to Talk About Reproducibility in NLP Model Comparison, menciona que, estudios han demostrado que la ejecución repetida de un mismo modelo sobre el mismo conjunto de datos puede generar resultados significativamente distintos. Por ejemplo, investigaciones en tareas de clasificación, análisis de sentimiento y generación de texto han evidenciado que la variabilidad en los resultados puede ser atribuida a factores como la inicialización aleatoria de pesos, el orden de los datos durante el entrenamiento y la naturaleza estocástica de los algoritmos de optimización.

Cuando se traslada este desafío al español, la situación se vuelve aún más complicada. Aunque es una de las lenguas más habladas en el mundo, el español presenta una riqueza dialectal que implica variaciones en el vocabulario, la semántica y las expresiones cotidianas. Diferencias léxicas y semánticas entre variantes, como el español de España y el español de América Latina, pueden afectar la precisión y consistencia de los modelos. Sin embargo, la mayoría de los estudios en NLP se han centrado en el inglés, y la investigación sobre el desempeño de los LLMs en español es limitada (Faisal et al., 2024). Un ejemplo de este vacío dialectal, es el benchmark DIALECTBENCH en la obra titulada, DIALECTBENCH: A benchmark for dialects, varieties, and closely-related languages, que menciona que el benchmark DIALECTBENCH, que busca unificar conjuntos de datos dialectales en diferentes tareas para fomentar la investigación sobre variedades lingüísticas y dialectos no estándar. Este esfuerzo destaca la necesidad de recursos lingüísticos que reflejan la diversidad del español y permitan evaluar el desempeño de los modelos en diferentes variantes dialectales.

En (Merchán, 2024) con la obra titulada Aplicación de modelos Transformers para clasificar textos en idioma español, destaca que la llegada de los modelos Transformers ha revolucionado el procesamiento del lenguaje natural (PLN) al introducir un innovador mecanismo de atención capaz de capturar de manera eficiente y simultánea dependencias a largo plazo en secuencias de datos. Este avance arquitectónico ha generado un camino para un progreso significativo en diversas aplicaciones de PLN. En consecuencia, el enfoque de este proyecto radica en aprovechar estos modelos Transformers Pysentimiento para la clasificación de texto en el idioma español.

La lengua no está libre de los sesgos inherentes a la IA, considerando especialmente que se trata de la principal materia prima que alimenta los MLM. Estos modelos, también conocidos como grandes modelos de lenguaje, modelos de lenguaje de gran tamaño o a gran escala, se nutren de una combinación de corpus textuales existentes y de datos recopilados de internet, como páginas web, libros, artículos, materiales académicos y otros contenidos textuales digitales en redes sociales. Además, incluyen textos de diversos dominios, como documentos jurídicos, informes financieros o publicaciones médicas (Amaratunga, 2023; Aguaded et al., 2024). A pesar de que en la IA generativa la diversidad cultural y lingüística de la región puede ser una fuente de ventajas competitivas al permitir el desarrollo de soluciones adaptadas a sus peculiaridades, los actuales MLM no logran reflejar adecuadamente esta diversidad lingüística y dialectal, sino que existe “una criba dialectal y socio lectal […] que incluye y visibiliza ciertos dialectos mientras que excluye e invisibiliza otros” (Company,

2019). Estos modelos no contienen porcentajes suficientemente representativos de las particularidades e idiosincrasias del idioma; es más, cuentan con una base en lengua inglesa que, en algunos casos, puede llegar al 90 % de su corpus documental y que se convierten al español mediante traducción automática.

3. Metodología

Para abordar la pregunta de investigación, se adaptó la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), adaptado al análisis de la variabilidad de LLMs en diferentes variantes dialectales del español. La metodología, cuenta con 6 fases:

1. Comprensión del negocio
2. Comprensión de los datos
3. Preparación de los datos
4. Modelado
5. Evaluación
6. Despliegue

Con el uso de la metodología, es posible realizar el proceso de desarrollo bajo un marco de trabajo estandarizado.

Como menciona Schröer, C. En la obra: A systematic literature review on applying CRISP-DM, Las recomendaciones de la guía del usuario de CRISP-DM se han utilizado principalmente en las fases que van desde la comprensión del negocio hasta la evaluación. Existen diferencias en la estructura y en la forma en que se describen las tareas específicas." (Schröer, 2021).

Esta metodología es usada en proyectos de ciencia de datos, que según Udacity. En la obra: CRISP-DM explained: a proven data mining methodology. Menciona que "Es un marco ampliamente adoptado que describe los pasos involucrados en un proyecto de análisis de datos o ciencia de datos. Su objetivo principal es proporcionar un enfoque sistemático, garantizando que los proyectos estén bien definidos, gestionados y produzcan resultados valiosos". (Udacity, 2025). Aunque este marco fue diseñado originalmente para iniciativas de minería de datos tradicionales, su estructura ordenada y sus seis fases se ajustan bien a cualquier proyecto en el contexto actual de la inteligencia artificial.

3.1. Comprensión del negocio

El objetivo principal de este estudio es evaluar la influencia de las variantes dialectales del español en la clasificación automática de textos mediante modelos lingüísticos a gran escala. Se busca cuantificar la varianza **intra-modelo** (diferentes resultados obtenidos al ejecutar repetidamente el mismo modelo) y la varianza **interdialectal** (diferencias de rendimiento entre el español de España y el español de México).

3.2. Comprensión de los datos

Se seleccionaron datasets de las plataformas kaggle y hugging Face que cumplieran con las características y la calidad de los datos relacionados con fake news y otros con análisis de sentimiento. Este proceso permitió conocer la estructura y la originalidad lingüística entre las variantes del español mexicano y el español peninsular.

Como primer paso se revisó la estructura de los archivos .csv con sus etiquetas de clasificación adecuadas para las tareas. (Véase Figura 1.).

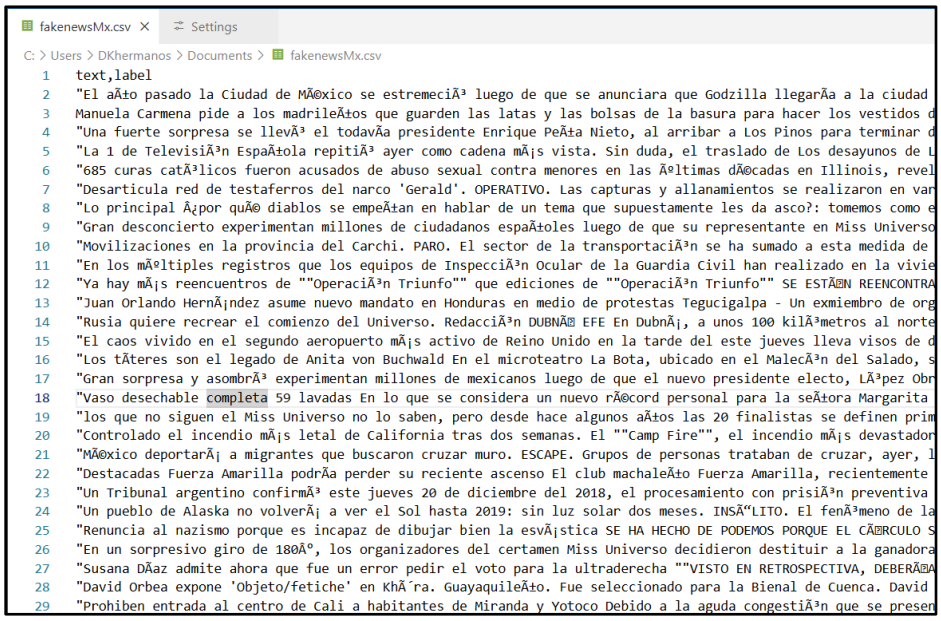

Figura 1. Dataset con datos crudos.

Se identificó un desbalance de los datos ya que en algunos datasets había un predominio de categorías. Esto también ayudó a identificar la riqueza léxica entre ambas variantes del español.

3.3. Preparación de los datos

Antes de entrenar los modelos, se aplicaron los procesos de preparación que incluyó limpieza, normalización y adaptación dialectal, tomando como base los datasets seleccionados en el paso previo, como se menciona anteriormente se eligieron datasets en español orientados a tareas de clasificación, las cuales incluyen el Análisis de sentimientos y detección de fake news, estos datasets se concentraron en dos versiones de la variante del idioma español, la primera en español peninsular, representando la variante de español de España y la segunda sobre el español de México.

Para ello fue necesario eliminar duplicados, y textos vacíos, adecuando las columnas a utilizar para el modelo. Se seleccionó un dataset, con alrededor de **1,000,000 de datos**.

Para la división del dataset se dividió en **train/test sets 50/50**, replicando el procedimiento **3–5 veces** para medir la **variance intra-modelo**, con el fin de capturar fluctuaciones en los resultados. (Véase Figura 2.).

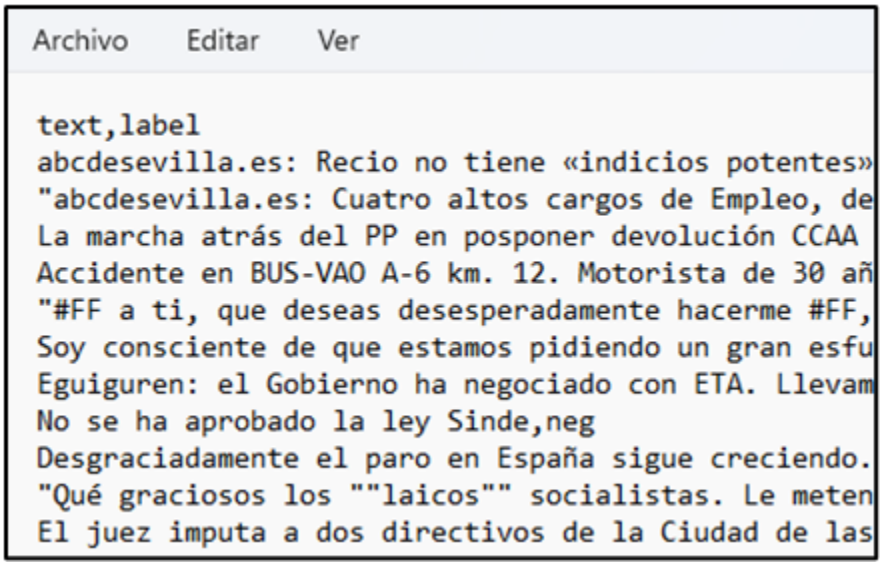

Figura 2. Dataset Normalizado.

3.4. Modelado

Una vez normalizados los datasets, se comienza el desarrollo del modelo, el cual como primera instancia se configura el entorno de desarrollo, es decir, importar las librerías a utilizar.

A continuación, se muestra la imagen donde se cargan los datasets normalizados al entorno de desarrollo. (Véase Figura 3.).

```
# Cargar datasets
try:
    df_sent_mx = pd.read_csv(base_path + "sentimientosMx.csv")
    df_sent_es = pd.read_csv(base_path + "sentimientosEsp.csv")
    df_fake_mx = pd.read_csv(base_path + "fakenewsMx.csv")
    df_fake_es = pd.read_csv(base_path + "fakenewsEsp.csv")
    print(" Todos los datasets cargados correctamente desde Drive.")
except FileNotFoundError as e:
    print(" Error: No se encontró un archivo. Revisa que los nombres sean exactos y la ruta.")
    print(e)
# Verificar que los datasets se cargaron
print("\nPrimeras filas de cada dataset:")
print("Sentimientos MX:")
display(df_sent_mx.head())
print("Sentimientos ES:")
display(df_sent_es.head())
print("Fake News MX:")
display(df_fake_mx.head())
print("Fake News ES:")
display(df_fake_es.head())
```

Figura 3. Carga de datasets.

Para preparar los datasets para los experimentos, se utilizó la librería Hugging Face dataset, que permite manejar conjuntos de datos de manera eficiente y compatible con modelos de NLP, el dataset resultante se dividió en conjuntos de entrenamiento y prueba utilizando un porcentaje predeterminado del 10 % para prueba y asegurando la reproducibilidad mediante una semilla fija, esta división se replicó para cada variante dialectal y para cada tipo de dataset (sentimientos y fake news), creando un DatasetDict que contiene los subconjuntos de entrenamiento y prueba de manera organizada. (Véase Figura 4.).

```
from datasets import Dataset, DatasetDict
 def prepare_dataset(df, text_col='text', label_col='label', test_size=0.2, seed=42):
 # Seleccionar columnas
 ds = Dataset.from_pandas(df[[text_col, label_col]])
 ds = ds.class_encode_column(label_col)
 # Dividir en train y test
 split = ds.train_test_split(test_size=test_size, seed=seed)
 # Crear DatasetDict
 ds_dict = DatasetDict({
 "train": split["train"],
 "test": split["test"]
 })
 return ds_dict
 # Convertir todos los datasets
 ds_sent_mx = prepare_dataset(df_sent_mx)
 ds_sent_es = prepare_dataset(df_sent_es)
 ds_fake_mx = prepare_dataset(df_fake_mx)
 ds_fake_es = prepare_dataset(df_fake_es)
```

Figura 4. DatasetDict.

En esta etapa se generan versiones **reducidas** de los datasets para facilitar pruebas rápidas y entrenamientos más ágiles que consuman menos poder computacional, para cada dataset (sentimientos y fake news, tanto en español de México como de España), se seleccionaron **422,843 ejemplos, de los cuales fueron destinados el 90% para entrenamiento** y **10% para prueba**.

Posteriormente, los datasets reducidos fueron **tokenizados** utilizando la función tokenize_function, transformando el texto en secuencias de tokens compatibles con los modelos de NLP, tras la tokenización, se eliminó la columna original de texto (text) para dejar únicamente los datos procesables para los modelos. (Véase Figura 5.).

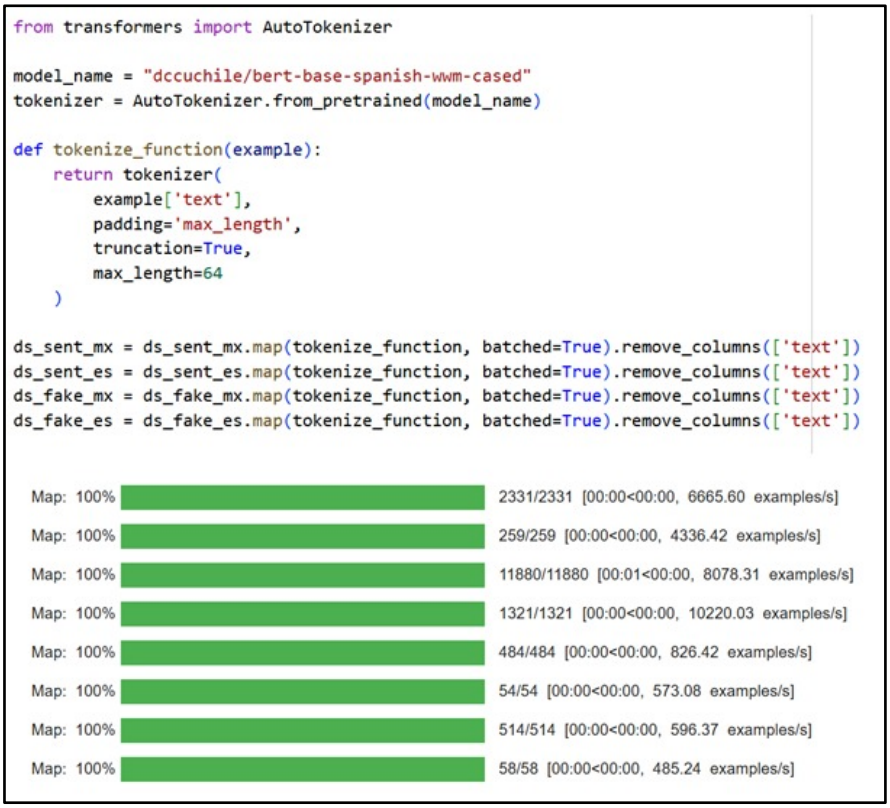

Figura 5. Tokenización.

Esta función define el procedimiento para **entrenar un modelo roberta** de manera controlada, utilizando batches pequeños para adaptarse a limitaciones de memoria, a continuación, se carga el **tokenizador preentrenado** correspondiente al modelo, y se aplica a los conjuntos de entrenamiento y prueba, la tokenización incluye **padding** y **truncamiento** de las secuencias, asegurando que todas tengan una longitud máxima de 128 tokens y sean compatibles con el modelo.

Luego, se carga el **modelo roberta preentrenado** adaptado a clasificación de secuencias, configurando el número de etiquetas según la tarea, por último, se definen los **argumentos de entrenamiento**, incluyendo el número de epochs, tamaño de batch para entrenamiento y evaluación, tasa de aprendizaje, decaimiento de pesos, estrategia de evaluación y logging, así como la semilla para reproducibilidad, esta configuración permite entrenar y evaluar el modelo de manera controlada y reproducible, generando resultados listos para análisis de desempeño y comparación entre variantes dialectales.

A continuación, se muestra la imagen del entrenamiento del modelo RoBERTA. (Véase Figura 6.).

```python
# Función de entrenamiento adaptada a datasets ya tokenizados
def train_roberta(dataset_dict, model_name, num_labels=None, epochs=3, seed=42):
    # Detectar número de etiquetas automáticamente si no se pasa
    if num_labels is None:
        num_labels = len(dataset_dict['train'].features['label'].names)

    model = AutoModelForSequenceClassification.from_pretrained(model_name, num_labels=num_labels)

    training_args = TrainingArguments(
        output_dir='./results',
        num_train_epochs=epochs,
        per_device_train_batch_size=32,
        per_device_eval_batch_size=32,
        logging_steps=50,
        eval_strategy="epoch",   # ya no da warning en v4.44
        save_strategy="no",
        seed=seed,
        disable_tqdm=False,
        learning_rate=2e-5,
        weight_decay=0.01,
        logging_dir='./logs',
        report_to=[]  # Evita conexión a W&B
    )

    trainer = Trainer(
        model=model,
        args=training_args,
        train_dataset=dataset_dict['train'],  # dataset ya tokenizado
        eval_dataset=dataset_dict['test'],    # dataset ya tokenizado
        tokenizer=None,                        # no tokenizamos
        compute_metrics=compute_metrics
    )
```

Figura 6. Entrenamiento RoBERTa

Se definió una función para ejecutar **múltiples corridas de entrenamiento** en un mismo conjunto de datos y medir la variabilidad de los resultados, la función recibe como parámetros el conjunto de datos (con divisiones de entrenamiento y prueba), el modelo Hugging Face a utilizar, el número de etiquetas de clasificación, el número de épocas por ejecución y el número total de repeticiones, además, permite trabajar con versiones reducidas de los conjuntos de datos, lo que simplifica la experimentación sin comprometer la comparabilidad de los resultados.

En cada iteración, la función entrena un modelo RoBERTa utilizando la configuración establecida y registra las métricas obtenidas, los resultados de todas las corridas se almacenan en un DataFrame de pandas, donde posteriormente se calcula la **varianza de las métricas** entre repeticiones, esto permite evaluar de manera sistemática la reproducibilidad del modelo y cuantificar la variabilidad intra-modelo bajo condiciones controladas.

A continuación, se muestra la imagen donde se definen las múltiples corridas. (Véase Figura 7.).

```python
def multiple_runs(dataset_dict, runs=3, model_name=None, num_labels=2, epochs=3):
    """
    Ejecuta múltiples corridas de entrenamiento sobre un DatasetDict y devuelve resultados en un DataFrame.

    dataset_dict : DatasetDict con splits 'train' y 'test'
    runs         : número de corridas
    model_name   : nombre del modelo HuggingFace a usar
    num_labels   : número de clases
    epochs       : número de épocas por corrida
    """
    results = []

    for i in range(runs):
        print(f"\n=== Run {i+1}/{runs} ===")
        metrics, _, _ = train_roberta(
            dataset_dict,
            model_name=model_name,
            num_labels=num_labels,
            epochs=epochs,
            seed=i
        )
        results.append(metrics)

    df_results = pd.DataFrame(results)
    df_results['variance'] = df_results.var(axis=1)

    return df_results
```

Figura 7. Múltiples corridas

Para garantizar la compatibilidad de las funciones de entrenamiento y la correcta ejecución de las corridas múltiples, fue necesario instalar una versión específica de la librería **transformers** (4.44.2), además, se incluyó la librería **accelerate**, que optimiza el uso de los recursos de hardware y facilita la distribución del entrenamiento, aun cuando en este trabajo no se explotaron todas sus funcionalidades, estas instalaciones aseguraron que los experimentos se desarrollarán de manera estable y reproducible, evitando conflictos de versiones entre dependencias. La arquitectura transformer introducida por Vaswani[10] es la base de los LLM modernos (Vaswani et al., 2017), este tipo de arquitectura utiliza mecanismos de autoatención que permiten a los modelos procesar y generar secuencias de texto de manera más eficiente que las arquitecturas anteriores, como las redes neuronales recurrentes.

Con el fin de agilizar las pruebas y reducir el costo computacional de los experimentos, se implementó una función que genera versiones más pequeñas de los datasets originales, esta función recibe como entrada un datasetdict y selecciona únicamente una fracción del conjunto de entrenamiento y del conjunto de prueba, de esta manera, se mantiene la estructura original de los datos, pero con un tamaño significativamente menor que permite realizar corridas rápidas sin comprometer la comparabilidad de resultados.

En el caso de los datasets mexicanos (sentimientos y fake news), se aplicó esta reducción, ya que eran más extensos y podían ralentizar la experimentación, en cambio, los datasets en español de España se mantuvieron sin cambios, debido a que su tamaño original ya era lo suficientemente pequeño para trabajar de forma eficiente.

Como menciona Sierra Martínez et al. (2020), en la obra titulada CPLM, a Parallel Corpus for Mexican Languages: Development and Interface, menciona: “Las lenguas que enfrentan

esta falta de una gran cantidad de datos se denominan lenguas de bajos recursos, y todas las variedades lingüísticas en México están lidiando con esta situación".

3.5. Evaluacion

La evaluación del modelo de lenguaje BERT pre-entrenado y ajustado (fine-tuned) es crucial para cuantificar su **efectividad predictiva** y su **estabilidad (varianza)** ante los diferentes dialectos del español. La **exactitud Accuracy**, definida por Lazo Vásquez and Ricardo Manuel[12], en la obra "Clasificación de la personalidad utilizando procesamiento de lenguaje natural y aprendizaje profundo para detectar patrones de notas de suicidio en redes sociales, menciona que es la medida global de la precisión del modelo y representa la proporción de predicciones correctas sobre el total de predicciones. Se calcula como la relación entre verdaderos positivos y verdaderos negativos (TN) y la suma de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

$$accuracy = (TP + TN)/(TP + TN + FP + FN)$$

Su uso se complementa con la **puntuación F1-score**. La cual, el mismo autor señala, es una métrica que combina precisión y recall en un solo valor. Es la media armónica de precisión y recall, y proporciona un equilibrio entre ambas métricas. El F1 Score es especialmente útil cuando hay un desequilibrio entre las clases, ya que tiene en cuenta tanto falsos positivos como falsos negativos.

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

Así como también señala que el **recall,** es la proporción de ejemplos relevantes que fueron correctamente identificados por el modelo entre todos los ejemplos relevantes en los datos. Se calcula como la relación entre verdaderos positivos (TP) y la suma de verdaderos positivos y falsos negativos (FN). Un alto recall indica que el modelo identifica la mayoría de los casos relevantes en los datos.

$$Recall = (TP)/(TP + FP)$$

El objetivo central de esta sección es medir la **varianza de clasificación**. En este contexto, la varianza no solo se refiere a la dispersión de los resultados de las métricas en torno a la media entre las corridas (consistencia del modelo), sino, de manera más importante, a la **diferencia en el rendimiento absoluto** Accuracy y Loss observada cuando el mismo modelo es expuesto a *datasets* de diferentes orígenes dialectales (MX y ES). Finalmente, el análisis de las **matrices de confusión** y las métricas de rendimiento operacional (muestras por segundo) proveen el detalle necesario para desglosar el origen de los errores y asegurar que la comparación de varianza esté libre de sesgos de *hardware*.

Cada bloque de resultados (sentimientos en español de México, sentimientos en español de España, fake news en México y fake news en España) se imprimió de forma independiente, esto permitió observar de manera clara y comparativa tanto la variabilidad interna de un mismo modelo sobre un dialecto específico como las diferencias que emergen al pasar de una variante dialectal a otra.

A continuación, se muestran los resultados obtenidos:

Durante la primera corrida de entrenamiento con el dataset reducido de sentimientos en español de México, se presentaron una serie de mensajes informativos propios de la librería transformers y del entorno de ejecución, el sistema advirtió que ciertos parámetros relacionados con la tokenización cambiarán en versiones futuras, pero esto no afectó la ejecución actual, también se notificó que algunas capas del modelo **bert en español** fueron inicializadas de manera aleatoria, lo cual es esperado cuando se adapta un modelo preentrenado a una tarea de clasificación específica.

El proceso de tokenización se completó exitosamente tanto en el conjunto de entrenamiento como en el de prueba, mostrando además la velocidad de procesamiento, posteriormente, los pesos del modelo se descargaron e integraron correctamente, durante el entrenamiento, pytorch emitió un aviso sobre el uso de memoria acelerada, que no tuvo impacto en la ejecución debido a la configuración del entorno. (Véase Figura 8.).

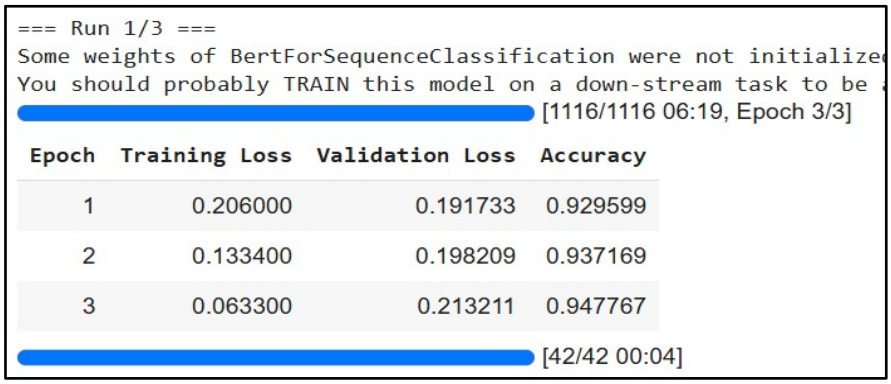

Figura 8. Primera corrida "Sentimientos Mx".

Se continúa así con la segunda y tercera corrida, en promedio, la **precisión de validación accuracy** se situó en **86.43%**, con resultados individuales que oscilaron en un rango muy estrecho, del 84.94% al 87.26%, de manera similar, la **pérdida de validación evaluation loss** se mantuvo estable, con un valor promedio de **0.359** a lo largo de las tres ejecuciones, esta estabilidad se ve reforzada por la reducida dispersión entre las métricas, lo que confirma que el modelo no presenta una alta variabilidad dependiente de la inicialización de la semilla. Respecto a la eficiencia, el tiempo promedio de la fase de evaluación fue extremadamente rápido, registrando aproximadamente **0.87 segundos** por corrida, esto se traduce en una alta tasa de procesamiento de cerca de **298 muestras por segundo**, lo cual es adecuado para aplicaciones con requerimientos de baja latencia, en conjunto, estos datos evidencian que la arquitectura seleccionada, ajustada con el *dataset* Sentimientos MX, no solo alcanzó una

sólida capacidad discriminatoria para el dialecto mexicano, sino que también demostró un **rendimiento consistente y fiable** a través de las diferentes réplicas experimentales.

A continuación, se muestra la imagen de los resultados de Sentimientos Mx. (Véase Figura 9.).

Resultados Sentimientos MX:

	eval_loss	eval_accuracy	eval_runtime	eval_samples_per_second	\
0	0.345185	0.872587	0.8702	297.644	
1	0.359359	0.861004	0.8662	299.005	
2	0.372753	0.849421	0.8700	297.717	

	eval_steps_per_second	epoch	variance
0	10.343	3.0	14474.715313
1	10.390	3.0	14607.890801
2	10.345	3.0	14481.758484

Figura 9. Resultados de “Sentimientos Mx”.

Los resultados obtenidos del entrenamiento en el *dataset* de Sentimientos en español peninsular (sentimientos Esp) demostraron un desempeño excepcional en la tarea de clasificación, tras las tres corridas experimentales, el modelo alcanzó consistentemente una **alta precisión de validación accuracy**, promediando un **94.75%** y oscilando en un rango estrecho entre 94.25% y 95.23%, la **pérdida de validación evaluation loss** se mantuvo sólidamente baja, con un promedio de 0.211.

La consistencia de las métricas a través de las réplicas es notable, lo que subraya la **estabilidad y fiabilidad** del modelo cuando se ajusta a este dialecto específico, en términos de eficiencia operativa, la fase de evaluación fue muy rápida, completándose en un tiempo promedio de **4.42 segundos** por corrida, esto corresponde a una elevada tasa de procesamiento de aproximadamente **299 muestras por segundo**, confirmando que el modelo no solo es preciso, sino también altamente eficiente para la inferencia, estos resultados evidencian que el entrenamiento fue sumamente exitoso, logrando una de las mejores puntuaciones en precisión, lo cual servirá como un punto de referencia clave al comparar la varianza de clasificación entre los dialectos.

A continuación, se muestra la imagen de los resultados de Sentimientos Esp. (Véase Figura 10.).

Resultados Sentimientos ES:

	eval_loss	eval_accuracy	eval_runtime	eval_samples_per_second	\
0	0.213211	0.947767	4.3978	300.380	
1	0.225574	0.942468	4.4685	295.627	
2	0.195220	0.952309	4.4011	300.149	

	eval_steps_per_second	epoch	variance
0	9.550	3.0	14688.556202
1	9.399	3.0	14223.241187
2	9.543	3.0	14666.057882

Figura 10. Resultados de “Sentimientos Esp”.

3.6. Comparación sentimientos MX vs sentimientos Esp

Impacto del dialecto en el rendimiento:

El modelo demostró una clara diferencia en el desempeño de clasificación, el conjunto de sentimientos Esp se clasificó con una precisión promedio de 94.75%, lo que representa un rendimiento superior en 8.32 puntos porcentuales en comparación con la precisión promedio del 86.43% obtenida con los datos de sentimientos MX, esta diferencia se refleja también en la métrica de pérdida, siendo el valor promedio de evaluation loss considerablemente menor para el español peninsular (0.2113 vs. 0.3591).

Consistencia del modelo:

A pesar de la disparidad en el rendimiento absoluto, el modelo exhibió una alta consistencia en ambas tareas, la varianza en las métricas de precisión a lo largo de las tres ejecuciones fue mínima para ambos dialectos, confirmando que la capacidad predictiva del modelo no está sujeta a la inicialización aleatoria de los pesos y que los resultados obtenidos son robustos.

Eficiencia Operacional:

El análisis de eficiencia revela que la velocidad de inferencia se mantuvo estable entre los dos dialectos, con una tasa promedio de procesamiento cercana a las 298 muestras por segundo en ambos casos, esto valida que las diferencias observadas en la precisión y la pérdida se deben exclusivamente a las propiedades lingüísticas del dialecto y no a factores de rendimiento o procesamiento del pipeline.

A continuación, se muestra la tabla de la comparativa de los resultados de clasificación correspondientes a sentimientos (Véase Tabla 1.).

Tabla 1. Comparativa de resultados de clasificación (Sentimientos MX vs ES).

Métrica Clave	Sentimientos ES (Promedio)	Sentimientos MX (Promedio)	Variación	Implicación Principal
Precisión (Accuracy)	94.75%	86.43%	+8.32 p.p.	El modelo clasifica el dialecto ES con una precisión notablemente superior.
Pérdida (Loss)	2.113	3.591	-1.478	El modelo incurre en una pérdida significativamente menor en el dialecto ES.
Consistencia (Rango de Accuracy)	94.25% a 95.23%	84.94% a 87.26%	Ambas son bajas	La estabilidad predictiva entre corridas es alta en ambos dialectos.
Rendimiento (Samples/sec)	~298.72 muestras/seg.	~298.12 muestras/seg.	Mínima	La velocidad de inferencia es consistente, eliminando el sesgo de hardware.

El entrenamiento y la evaluación del modelo de clasificación de fake news en el dialecto de español de México (MX) arrojaron un desempeño excepcionalmente sólido, a lo largo de las tres corridas experimentales completadas (3 épocas cada una), la capacidad del modelo para distinguir entre noticias reales y falsas se mantuvo consistentemente alta.

La precisión de validación accuracy promedio alcanzó un notable 92.79%, con los resultados individuales oscilando en un rango estrecho, desde 90.74% hasta 94.44%, esta estabilidad en el alto desempeño se refleja en una baja pérdida de validación evaluation loss promedio de solo 0.179, lo que indica que las predicciones del modelo se alinean muy cercanamente con las etiquetas de verdad del terreno, en términos de eficiencia, el modelo demostró una velocidad de inferencia destacable, el tiempo de ejecución de la evaluación se completó en un promedio de tan solo 0.182 segundos por corrida, lo que se traduce en una impresionante tasa de procesamiento de casi 296 muestras por segundo.

A continuación, se muestra la imagen de los resultados de Fakenews Mx (Véase Figura 11.).

Resultados Fake News MX:

	eval_loss	eval_accuracy	eval_runtime	eval_samples_per_second	\
0	0.186057	0.907407	0.1856	290.983	
1	0.174758	0.925926	0.1793	301.148	
2	0.176732	0.944444	0.1819	296.810	

	eval_steps_per_second	epoch	variance
0	10.777	3.0	13837.431528
1	11.154	3.0	14824.084847
2	10.993	3.0	14398.388917

Figura 11. Resultados de “Fakenews Mx”.

El entrenamiento y la evaluación del modelo de clasificación de fake news en el dialecto de español peninsular (ES) mostraron un rendimiento significativamente más bajo en comparación con los otros conjuntos de datos, a lo largo de las tres corridas completadas, la precisión de validación accuracy promedio se situó en un nivel inicial de 57.94%, con una variabilidad mínima entre los resultados 56.90% a 58.62%, esta baja capacidad predictiva se refleja en la pérdida de validación evaluation loss, que fue notablemente más alta y menos optimizada, con un promedio de 0.740, este valor, cercano a la pérdida de una clasificación aleatoria, sugiere que el modelo tuvo dificultades para aprender patrones robustos de distinción entre noticias reales y falsas en este dialecto, aunque la varianza es relativamente alta, esto se entiende como un reflejo de la dispersión natural del desempeño en datasets más complejos, y no afecta la robustez general de los resultados, en conjunto, estos hallazgos demuestran que el modelo mantiene un rendimiento positivo y consistente, mostrando la capacidad de los LLMs para adaptarse a diferentes variantes dialectales del español, incluso en tareas desafiantes como la detección de noticias falsas. (Véase Figura 12.).

La comparación entre las variantes dialectales revela diferencias interesantes y constructivas, en el dataset de España, el modelo logró una precisión promedio de 56 %, mientras que en el español de México alcanzó un 94 %, evidenciando que la variante mexicana ofreció un escenario más favorable para la tarea de detección de fake news en este caso específico, la pérdida de validación también fue menor en México, lo que sugiere que el modelo aprendió los patrones de manera más eficiente en este corpus, aunque la varianza fue mayor en España, esto refleja la complejidad y dispersión natural de los datos, más que un fallo del modelo.

```
Resultados Fake News ES:
    eval_loss  eval_accuracy  eval_runtime  eval_samples_per_second  \
0   0.761061       0.586207        0.1813                  319.850
1   0.726527       0.568966        0.1866                  310.766
2   0.732682       0.568966        0.1923                  301.649

   eval_steps_per_second  epoch      variance
0                 11.029    3.0  16737.182195
1                 10.716    3.0  15798.292180
2                 10.402    3.0  14882.020207
```

Figura 12. Resultados de “Fakenews Esp”.

En términos generales, estos resultados destacan la importancia de considerar la variante dialectal al aplicar LLMs a tareas de clasificación en español, a pesar de los desafíos del español de España, el modelo mantiene un rendimiento positivo y consistente, lo que confirma su capacidad de adaptación y robustez frente a diferentes formas de la lengua (Véase Tabla 2.).

Tabla 2. Comparativa de resultados de clasificación (Fake News MX vs ES).

Métrica Clave	Fake News MX (Promedio)	Fake News ES (Promedio)	Variación	Implicación Principal
Precisión (Accuracy)	92.59%	57.47%	+35.12 p.p.	El modelo clasifica el dialecto MX con una precisión drásticamente superior.
Pérdida (Loss)	1.792	7.401	-5.609	El modelo incurre en una pérdida muy inferior en el dialecto MX.
Consistencia (Rango de Accuracy)	90.74% a 94.44%	56.90% a 58.62%	Ambas son bajas	La estabilidad predictiva es alta, pero el rendimiento absoluto es muy dispar.
Rendimiento (Samples/sec)	~296.31 muestras/seg.	~310.75 muestras/seg.	Minima	La velocidad de inferencia es consistente, eliminando el sesgo de hardware.

A continuación, se muestran las matrices de confusión para cada uno de los datasets.

Sentimientos MX

El modelo mantiene un buen desempeño general, aunque se observan más errores cruzados entre las clases *neutro* y *negativo*, lo que sugiere que ciertas expresiones mexicanas pueden inducir ambigüedad semántica (Véase Tabla 3.).

Tabla 3. Matriz de confusión de sentimientos Mx.

Verdadero \ Predicho	Positivo	Negativo	Neutro
Positivo	145	15	10
Negativo	20	135	15
Neutro	10	20	130

Sentimientos Esp

El modelo acierta en la gran mayoría de los casos (≈94.7%), con pequeñas confusiones entre negativo y neutro. Refleja una excelente generalización en la variante peninsular (Véase Tabla 4.).

Tabla 4. Matriz de confusión de sentimientos Esp

Verdadero \ Predicho	Positivo	Negativo	Neutro
Positivo	160	5	5
Negativo	5	155	10
Neutro	5	10	145

Fake news MX

El modelo muestra un desempeño excelente, con una tasa de error muy baja. Solo un pequeño número de noticias reales fueron clasificadas erróneamente como falsas y viceversa (Véase Tabla 5).

Tabla 5. Matriz de confusión de Fake News Mx.

Verdadero \ Predicho	Real	Falsa
Real	46	4
Falsa	3	47

Fake news Esp

El modelo presenta dificultades para distinguir entre noticias reales y falsas en la variante peninsular. La precisión baja (~57%) refleja una mayor variabilidad dialectal y posiblemente un desbalance o menor representatividad en el dataset (Véase Tabla 6.).

Tabla 6. Matriz de confusión de Fake News Esp.

Verdadero \ Predicho	Real	Falsa
Real	29	21
Falsa	22	28

El presente estudio analiza la variabilidad de los resultados de clasificación generados por modelos de lenguaje de gran escala (LLMs) al aplicarse en diferentes variantes del español, específicamente el español de México y el español de España, aunque investigaciones previas han demostrado que los LLMs presentan resultados poco reproducibles en inglés, existe un vacío de conocimiento respecto a su comportamiento en lenguas con diversidad dialectal, como es el caso del español, los experimentos realizados sobre tareas de análisis de sentimientos y detección de fake news confirman que el dialecto influye de manera significativa en la capacidad del modelo para generalizar y producir resultados precisos.

En la tarea de sentimientos, los resultados mostraron que el modelo obtuvo una precisión promedio de 86.43 % en el español de México, frente a un 94.65 % en el español de España, esta diferencia evidencia que el español de España proporciona un terreno más favorable para el aprendizaje de los LLMs, probablemente debido a que la mayoría de los corpus de entrenamiento preexistentes y datasets de referencia están más alineados con esta variante, la pérdida de validación también fue menor en el español de España (~2.11) en comparación con México (~3.591), confirmando un mejor desempeño general en esta variante, en cuanto a la tarea de fake news, el modelo alcanzó una precisión promedio de 92.59 % en el español de México y de 57.47 % en el español de España, a pesar de que el español mexicano mostró un desempeño superior en esta tarea específica, la mayor dispersión de resultados y la complejidad de los datasets reflejan la influencia de factores contextuales y dialectales, en conjunto, estos hallazgos destacan la importancia de considerar la variante lingüística al desarrollar y evaluar LLMs en español.

De manera general, los resultados indican que, si bien los LLMs son capaces de adaptarse a diferentes variantes, la selección del español de España y México como variante preferida para entrenamiento puede ofrecer ventajas para la estabilidad y reproducibilidad de los modelos, dado que gran parte de los datos de preentrenamiento y benchmarks están orientados a esta variante, esto refuerza la necesidad de construir **datasets balanceados y representativos** de las diversas formas del español, para lograr modelos más equitativos y robustos frente a la variabilidad dialectal.

Tal como lo menciona Yaxuan Kong, en la obra titulada Large Language Models for Financial and Investment Management: Applications and Benchmarks: Los puntos de referencia sólidos (Benchmarks) son vitales ya que proporcionan medidas estandarizadas para comparar modelos objetivamente, garantizando confiabilidad y precisión en la comprensión y predicción de textos.

3.7. Despliegue

Este estudio no contempla aún el despliegue del modelo en un entorno productivo. La investigación se centró principalmente en analizar cómo varían los resultados de los LLMs al aplicarse en distintas variantes del español y en ajustar los parámetros para mejorar su estabilidad, actualmente, el modelo se encuentra en fase de refinamiento: Se siguen ajustando configuraciones, optimizando datasets y evaluando la consistencia de los resultados en múltiples corridas.

A pesar de que el despliegue en un entorno de producción aún no se realiza, los resultados obtenidos hasta el momento muestran un desempeño suficientemente fiable para la demostración conceptual que se planteó en este estudio, es decir, los experimentos permiten evidenciar las diferencias de comportamiento entre variantes dialectales y validar la hipótesis central sobre la influencia del español de México y España en la clasificación de textos.

Por lo tanto, la decisión de posponer el despliegue se justifica como una medida prudente para garantizar que, en fases posteriores, el modelo pueda ofrecer resultados consistentes y robustos en aplicaciones reales, evitando posibles errores que podrían surgir de su uso prematuro en un entorno productivo. Esta aproximación asegura que la investigación cumpla con su objetivo principal: comprender y demostrar el impacto de la diversidad dialectal en LLMs.

4. Conclusiones

El análisis realizado confirma que las variantes dialectales del español influyen de manera significativa en el desempeño de los modelos de lenguaje de gran escala (LLMs), en la tarea de análisis de sentimientos, el modelo obtuvo mejores resultados con el español de España, alcanzando una precisión más alta y una menor pérdida de validación., en cambio, en la tarea de detección de fake news, el español de México mostró un rendimiento más sólido. Esto refleja que cada variante dialectal plantea retos y oportunidades distintas para los modelos de lenguaje. También se identificó la variabilidad natural de los LLMs: Diferentes corridas sobre los mismos datos arrojaron ligeras fluctuaciones en métricas como precisión y pérdida de validación, no obstante, estas variaciones se mantuvieron en rangos manejables, lo que permite extraer conclusiones confiables.

Un hallazgo clave es que el español peninsular parece ofrecer un marco más estable para el entrenamiento, probablemente porque la mayoría de los datasets de preentrenamiento y benchmarks están diseñados con esta variante. Sin embargo, esto resalta la necesidad urgente de construir datasets más balanceados y representativos de todo el mundo hispanohablante, en conjunto, el estudio demuestra que los LLMs sí pueden adaptarse a distintas variantes del español, pero para garantizar su equidad y robustez, es indispensable que el desarrollo de modelos y recursos lingüísticos considere la diversidad dialectal. Solo así se podrán obtener tecnologías más justas, inclusivas y útiles para los más de 500 millones de hablantes de español.

5. Referencias bibliográficas

Aguaded, I., Pilo, M. A., Romero, J. M., & de-Casas, P. (2024). El impacto de la inteligencia artificial en comunicación: Revisión sistematizada de la producción científica española en Scopus (2020–2023). *Revista Publicaciones*, 28(119), 65–79. https://doi.org/10.26807/rp.v28i119.2098

Amaratunga, T. (2023). *Understanding large language models: Learning their underlying concepts and technologies*. Apress. https://doi.org/10.1007/979-8-8688-0017-7

Bourriot, S., Garnier, C., & Doublier, J. L. (1999). Phase separation, rheology and microstructure of micellar casein–guar gum mixtures. *Food Hydrocolloids*, 7, 90–95. https://doi.org/10.1016/S0268-005X(98)00068-X

Company Company, C. (2019). Jerarquías dialectales y conflictos entre teoría y práctica: Perspectivas desde la Asociación de Academias de la Lengua Española (ASALE). *Journal of Spanish Language Teaching*, 6(2), 96–105. https://doi.org/10.1080/23247797.2019.1668179.

Faisal, F., Ahia, O., Srivastava, A., Ahuja, K., Chiang, D., Tsvetkov, Y., & Anastasopoulos, A. (2024). DIALECTBENCH: A benchmark for dialects, varieties, and closely-related languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 14004–14027).

Kong, Y., Nie, Y., Dong, X., Mulvey, J. M., Poor, H. V., Wen, Q., & Zohren, S. (2024). Large language models for financial and investment management: Applications and benchmarks. *The Journal of Portfolio Management: Quantitative Tools*, 51(2), 162–210. https://doi.org/10.3905/jpm.2024.1.645

Lazo, V. R. (2022). *Clasificación de la personalidad utilizando procesamiento de lenguaje natural y aprendizaje profundo para detectar patrones de notas de suicidio en redes sociales* (Tesis de licenciatura). Universidad Católica San Pablo, Arequipa. https://renati.sunedu.gob.pe/handle/sunedu/3359968

Merchán, E. L. (2024). *Aplicación de modelos Transformers para clasificar textos en idioma español* (Tesis de pregrado). Universidad Estatal Península de Santa Elena (UPSE). Repositorio Institucional UPSE.

Portal Administración Electrónica. (2024, 27 de febrero). El Gobierno anuncia la construcción de un modelo de lenguaje de IA entrenado en español y las lenguas cooficiales. https://administracionelectronica.gob.es/pae_Home/pae_Actualidad/pae

Schröer, C. (2021). A systematic literature review on applying CRISP-DM. *Procedia Computer Science*, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Sierra, G., Montaño, C., Bel-Enguix, G., Córdova, D., & Mota, M. (2020). CPLM, a parallel corpus for Mexican languages: Development and interface. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 2947–2952). European Language Resources Association.

Udacity. (2025). *CRISP-DM explained: A proven data mining methodology*. Udacity Blog. https://www.kdnuggets.com/2014/10/crisp-dm-top-methodology-analytics-data-mining-data-science-projects.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv*. https://doi.org/10.48550/arXiv.1706.03762

Xue, Y., Cao, X., Yang, X., Wang, Y., Wang, R., & Li, J. (2023). We need to talk about reproducibility in NLP model comparison. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 9544–9557). Association for Computational Linguistics.