

Comparación de modelos Transformer para clasificar poesía en español

Comparing Transformer Models for Spanish Poetry Classification

J. C. Santana Becerril¹ , R. Alejo Eleuterio¹ *, S. M. Martínez Gallegos¹ , F. Del Razo López¹ , E. E. Granda

Gutiérrez² , G. Miranda Piña³

(*ralejoe@toluca.tecnm.mx)

Recibido: 06 de noviembre de 2025 Aceptado: 08 de enero de 2026

RESUMEN

La poesía en español, por su riqueza estilística y el uso de recursos retóricos, plantea retos particulares para la clasificación automática. Cualquier sesgo en etapas clave, incluyendo la representación del texto, la arquitectura del modelo o el protocolo de evaluación, puede impactar el desempeño; por ello, resulta pertinente comparar familias de modelos bajo condiciones homogéneas. En este artículo se estudia la efectividad relativa de cuatro arquitecturas basadas en *Transformers* ampliamente utilizadas: BERT-base uncased (DCC), BERT-base cased (DCC), RoBERTa-base (BERTIN) y DistilBERT uncased (DCC). La metodología emplea un protocolo único de ajuste fino para cada modelo (mismos hiperparámetros iniciales, partición proporcional por clase y control de semilla), con evaluación mediante métricas estándar (accuracy, precision, recall, F1) y análisis cualitativo de *embeddings* antes y después del entrenamiento. El objetivo es identificar, en un escenario de datos limitados, qué arquitectura ofrece el mejor equilibrio entre rendimiento y costo computacional, así como caracterizar errores típicos asociados al estilo poético. Los hallazgos buscan ofrecer lineamientos reproducibles para la selección de modelos en *corpus* pequeños de poesía en lengua española. **Palabras clave:** Transformers, Clasificación de texto, Poesía en español, BERT, RoBERTa, DistilBERT

¹División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/Instituto Tecnológico de Toluca, Av. Tecnológico s/n, Col. Agrícola, Metepec 52149, México.

²Centro Universitario UAEM Atlacomulco, Universidad Autónoma del Estado de México, Km. 60 Carretera Toluca–Atlacomulco, Atlacomulco 50450, México.

³División de Ingeniería en Sistemas Computacionales, Tecnológico de Estudios Superiores de Jocotitlán, Carretera Toluca–Atlacomulco km 44.8, Ejido de San Juan y San Agustín, Jocotitlán 50700, México.



ABSTRACT

Spanish poetry, with its stylistic richness and rhetorical devices, poses specific challenges for automatic classification. Any bias at key stages, including text representation, model architecture, or evaluation protocol, may affect performance; therefore, comparing model families under homogeneous conditions is pertinent. This article studies the relative effectiveness of four widely used *Transformer* architectures: BERT-base uncased (DCC), BERT-base cased (DCC), RoBERTa-base (BERTIN), and DistilBERT uncased (DCC). The methodology applies a single fine-tuning protocol (shared initial hyperparameters, stratified split, and seed control), evaluates with standard metrics (accuracy, precision, recall, F1), and includes a qualitative analysis of embeddings before and after training. The goal is to identify, in a limited-data scenario, which architecture offers the best balance between performance and computational cost, and to characterize typical errors linked to poetic style. The findings aim to provide reproducible guidelines for model selection on small Spanish poetry corpora. **Keywords:** Transformers, Text classification, Spanish poetry, BERT, RoBERTa, DistilBERT

INTRODUCCIÓN

La clasificación automática de poesía en español es relevante para el análisis literario, la curaduría digital y la recuperación temática de textos, pues permite identificar de forma sistemática estructuras estróficas y patrones métricos que complementan la labor del especialista humano y facilitan la gestión de grandes colecciones digitales [1, 2]. A diferencia de otros géneros, los poemas presentan alta variación formal, ambigüedad semántica y uso intensivo de recursos retóricos, lo que dificulta su representación y la separación consistente entre categorías como *amor* y *político-social*.

Una elección inadecuada del modelo o del protocolo de evaluación puede traducirse en bajo desempeño y conclusiones poco confiables, afectando la toma de decisiones sobre qué arquitectura aplicar en colecciones reales de poemas. Este problema se acentúa en escenarios con pocos datos y con distribuciones de clase altamente desbalanceadas, donde pequeñas variaciones en la muestra de entrenamiento pueden producir cambios notables en el rendimiento por clase y en la estabilidad de los modelos [3].

En este contexto, se plantea un estudio comparativo de cuatro arquitecturas *Transformer* ampliamente utilizadas: BERT-base uncased (DCC), BERT-base cased (DCC), RoBERTa-base (BERTIN) y DistilBERT uncased (DCC). El objetivo es identificar qué modelo ofrece el mejor equilibrio entre desempeño y costo computacional cuando el corpus es reducido y heterogéneo.

La metodología emplea un único protocolo de ajuste fino con hiperparámetros iniciales compartidos,

división de los datos que preserva la distribución por clase en entrenamiento, validación y prueba, y evaluación mediante métricas estándar (accuracy, precision, recall y F1 macro). Asimismo, se realiza un análisis cualitativo de *embeddings* antes y después del entrenamiento para explorar señales de separabilidad y patrones de error vinculados al estilo poético.

Este artículo se organiza de la siguiente manera: la Sección 2 presenta el estado del arte; la Sección 3 expone el marco teórico; la Sección 4 describe la metodología; la Sección 5 muestra los resultados; y la Sección 6 reúne las conclusiones y líneas futuras.

ESTADO DEL ARTE

La clasificación automática de textos literarios (en particular, de poesía) plantea retos adicionales frente a géneros informativos debido a la variación formal, la ambigüedad semántica y el uso intensivo de recursos retóricos. En este contexto, los modelos de lenguaje preentrenados basados en la arquitectura *Transformer* han desplazado progresivamente a enfoques utilizados previamente (SVM, CNN, LSTM) al ofrecer representaciones contextuales más ricas y transferibles entre tareas.

Para el español, la disponibilidad de modelos específicos ha sido determinante. BETO consolidó el uso de BERT preentrenado exclusivamente en español y mostró ventajas sobre baselines multilingües en múltiples tareas [4]. Posteriormente, la familia MarIA/BNE presentó modelos RoBERTa entrenados a gran escala con el corpus de la Biblioteca Nacional de España, ampliando sustancialmente la cobertura y calidad de los datos en español [5, 6]. En paralelo, variantes ligeras para español, como DistilBETO, buscaron reducir costo computacional manteniendo un rendimiento competitivo [7].

En dominios poéticos y cercanos, la evidencia reciente sugiere que los modelos basados en *Transformers* superan a arquitecturas anteriores cuando se dispone de un protocolo de ajuste fino controlado y datos curados. En un estudio con poemas de la *Poetry Foundation*, BERT alcanzó resultados de clasificación multicategoría que oscilaron por corpus y configuración, subrayando la sensibilidad a tokenización, longitud máxima de entrada y balanceo [8] de clases. Estos trabajos con BERT sobre poesía reportan mejoras frente a CNN/LSTM y discuten efectos de desbalance y tamaño de los conjuntos. De forma complementaria, en otras lenguas con recursos específicos (p. ej., árabe), AraBERT demostró el papel del preentrenamiento monolingüe en ganancias de precisión para clasificación y análisis de estilo [9].

Más allá del español, investigaciones en otras lenguas romances también han explorado la aplicación de modelos Transformer. En idiomas como el francés, el italiano y el portugués, el uso de modelos Transformer para la identificación y clasificación de textos ha mostrado resultados especialmente prometedores. En francés, modelos monolingües como CamemBERT han demostrado mejoras claras en tareas de clasificación de texto frente a modelos multilingües y métodos tradicionales [10], lo

que respalda la importancia de un preentrenamiento específico en la lengua para capturar matices léxicos y estilísticos propios. En italiano, trabajos recientes con variantes de BERT aplicadas a poesía han puesto de relieve su capacidad para manejar estructuras sintácticas poco canónicas y fenómenos léxico-semánticos propios del dominio literario [11], lo que abre la puerta a tareas de clasificación de estilo y época. De forma análoga, en portugués, modelos monolingües como BERTimbau se han empleado con éxito en la clasificación de poemas y de noticias [12], superando a representaciones basadas en word embeddings clásicos y evidenciando cómo factores como el desbalance de clases y la brevedad de los textos inciden en la estabilidad del entrenamiento y en la elección de métricas y estrategias de regularización.

MARCO TEÓRICO

La clasificación de textos es una aplicación central de la inteligencia artificial (IA) y el aprendizaje automático (ML), en la que se asigna a cada documento x una etiqueta y partiendo de datos etiquetados y una función de pérdida adecuada. Su evaluación se realiza con métricas estándar como exactitud, precisión, exhaustividad y F1, recomendándose reportes por clase cuando existen posibles desequilibrios o ambigüedades [13]. En este contexto, el aprendizaje profundo (DL) aporta representaciones distribuidas que capturan regularidades semánticas y sintácticas de manera más rica que los enfoques puramente contadores [14].

Representaciones del texto

Las representaciones estáticas (p. ej., word2vec, GloVe, fastText) asignan un vector fijo a cada palabra a partir de grandes corpus, con buenos resultados para relaciones léxicas generales, aunque limitadas para la polisemia y el contexto [15–17]. Las representaciones contextuales, habilitadas por arquitecturas modernas, generan vectores dependientes del entorno del token, lo que mejora sensiblemente tareas de clasificación, especialmente en textos breves y con ambigüedad como la poesía [18, 19].

Arquitectura Transformer y modelos preentrenados

La arquitectura *Transformer* introduce el mecanismo de autoatención para modelar dependencias de corto y largo alcance sin recurrencia ni convoluciones, con entrenamiento estable y paralelizable [18]. Sobre esta base, los modelos preentrenados (PLMs) aprenden representaciones contextuales en grandes corpus y se adaptan a tareas específicas mediante el ajuste fino [19]. Entre los más usados para clasificación están los encoders bidireccionales tipo BERT y sus variantes compactas (DistilBERT) [20, 21].

En español, destacan opciones monolingües como BETO [4] y familias entrenadas en grandes colecciones hispanas (BERTIN y RoBERTa-BNE) [5, 6, 22], así como alternativas multilingües (mBERT, XLM-R) [20, 23]. La evidencia empírica sugiere que, cuando el corpus objetivo es enteramente en

español, los modelos monolingües alinean mejor vocabulario y estadísticas del idioma; los multilingües resultan atractivos si se persigue transferencia entre lenguas o se dispone de datos escasos.

Tokenización para español

Los modelos PLM emplean segmentación por subpalabras para equilibrar cobertura y tamaño de vocabulario. WordPiece (BERT/BETO) y BPE a nivel de byte (RoBERTa/GPT-2) son los enfoques predominantes; SentencePiece-Unigram es otra alternativa extendida [24–28]. En español, preservar diacríticos, mayúsculas y signos propios (¿¡) suele ser beneficioso; en corpus poéticos, la fragmentación estable y el respeto a clíticos y contracciones (“dámelo”, “poniéndose”) ayudan a retener señales sutiles de tema y estilo. Por comparabilidad, se recomienda utilizar siempre el tokenizador nativo del modelo evaluado y reportar longitud media en tokens y porcentaje de truncado.

Protocolos de evaluación

Para realizar comparaciones justas, es aconsejable fijar particiones (entrenamiento/validación/prueba), semillas aleatorias y criterios de selección, así como reportar F1 macro junto con métricas globales cuando hay clases potencialmente desbalanceadas o de difícil separación [13]. En poesía, la separación a nivel de poema evita fugas por textos similares; cuando es posible, controlar por autor/obra reduce la sobrestimación asociada a estilo idiosincrático.

En este trabajo, adoptamos F1 macro como métrica principal para evaluar de forma equitativa el desempeño en ambas categorías, particularmente relevante dada la sensibilidad de la poesía a ambigüedades temáticas. Los detalles operativos del protocolo (particionado estratificado, fijación de semillas y criterios de selección del mejor modelo) se describen en la Sección 4 (Metodología).

Implicaciones para este estudio

La naturaleza breve, metafórica y a menudo ambigua de la poesía en español sugiere que los modelos con representaciones contextuales finas y buen manejo de subpalabras deberían aventajar a enfoques más simples. Esta hipótesis está respaldada por estudios recientes que muestran cómo las representaciones contextuales capturan mejor la polisemia, la ironía y las metáforas, que son rasgos centrales del lenguaje poético, en comparación con embeddings estáticos o modelos no contextuales [8, 19].

En este marco, comparar BERT-base (cased/uncased, DCC), RoBERTa-base (familias BERTIN/BNE) y variantes distilled (DistilBERT/DistilBETO) es pertinente por cuatro razones teóricas:

- **Alineación lingüística (monolingüe vs. multilingüe).** Los modelos entrenados específicamente en español tienden a capturar mejor diacríticos, clíticos y flexión, rasgos frecuentes y semánticamente relevantes en poesía.
- **Unidad básica de segmentación.** WordPiece (BERT/BETO) y BPE a nivel de byte (RoBERTa)

inducen patrones de fragmentación distintos; ello puede afectar qué señales temáticas quedan disponibles por verso/estrofa.

- **Capacidad vs. eficiencia.** Las variantes *distilled* reducen parámetros y costo, lo que plantea la pregunta conceptual de cuánto rendimiento temático se pierde (o conserva) en textos cortos y figurativos.
- **Estabilidad semántica en textos breves.** En poemas con señales dispersas, la robustez de las representaciones contextuales frente a polisemia, metáfora e intertextualidad es crucial para separar motivos como *amor* y *político-social*.

Diferencias entre los modelos evaluados

Los cuatro modelos comparados en este estudio comparten la arquitectura base de Transformer, pero difieren en aspectos clave de su preentrenamiento, tokenización y diseño, lo que puede influir en su desempeño en tareas poéticas. A continuación se resumen sus características distintivas:

- **BERT-base (DCC):** Entrenado en español sobre un corpus extenso del Common Crawl. Utiliza tokenización WordPiece y aplica *Whole Word Masking* (WWM), lo que mejora la coherencia semántica en palabras compuestas o con clíticos (p.ej., “dámelo”). Viene en versiones *cased* (preserva mayúsculas y acentos) y *uncased* (normaliza a minúsculas) [29]. La preservación de mayúsculas es relevante en poesía, donde el uso estratégico de mayúsculas iniciales o acentos puede indicar énfasis o ironía.
- **RoBERTa-base (BERTIN):** Variante de RoBERTa entrenada en español sobre un conjunto de datos aún más grande (incluyendo libros y prensa histórica). Usa tokenización BPE a nivel de byte, que maneja mejor caracteres fuera del vocabulario y es más robusta ante variaciones ortográficas [30]. A diferencia de BERT, no utiliza segmentación por oraciones (no necesita token [SEP]) y fue entrenado con mayor tamaño de lote y más pasos de optimización, lo que suele traducirse en representaciones más estables.
- **DistilBERT (DCC):** Versión destilada de BERT-base *uncased*, con 40 % menos de parámetros [31]. Conserva gran parte del rendimiento en tareas de clasificación al retener el 95 % de la capacidad de BERT en promedio, pero con menor capacidad para capturar matices finos en contextos altamente ambiguos o metafóricos, como los comunes en la poesía.

Estas diferencias (especialmente en tokenización, sensibilidad a la forma ortográfica y capacidad contextual) son relevantes en un dominio como la poesía, donde la forma y el significado están

estrechamente entrelazados. Asimismo, la pérdida de señales como mayúsculas o acentos en versiones *uncased* puede diluir pistas estilísticas críticas.

METODOLOGÍA

Este estudio sigue un protocolo experimental diseñado para comparar de forma justa el desempeño de distintas arquitecturas *Transformer* en la clasificación temática de poesía en español. El flujo general comprende: (1) selección y curación de un corpus binario balanceado de poemas de Pablo Neruda; (2) partición estratificada en conjuntos de entrenamiento, validación y prueba; (3) evaluación de cuatro modelos preentrenados en español bajo dos estrategias de adaptación (*feature-based* y *fine-tuning*) utilizando sus tokenizadores nativos y un conjunto común de hiperparámetros; (4) cuantificación del desempeño mediante métricas estándar (accuracy, precisión, recall, F1 macro) y matrices de confusión; y (5) análisis cualitativo de las representaciones latentes mediante proyecciones UMAP y revisión de errores.

A continuación se detalla cada componente del protocolo.

Conjunto de datos

Sea $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ con $N = 180$ poemas de Pablo Neruda. Cada x_i es el texto íntegro de un poema en español (versos y estrofas preservados) y el espacio de etiquetas es

$$\mathcal{Y} = \{\text{político-social, amor}\}.$$

La tarea es de *clasificación binaria*: dado x , el sistema estima $\hat{y} \in \mathcal{Y}$.

El conjunto de datos (180 poemas: 96 *Político-Social* y 84 *Amor*) tiene la forma `['text', 'label']`, donde `text` es el poema completo y `label` es la categoría como entero (0=*Amor*, 1=*Político-Social*).

Para el manejo del dataset se consideraron los siguientes factores: (1) Cobertura y trazabilidad: Se excluyen prólogos, notas editoriales, traducciones y duplicados/variantes mínimas. Se conservan metadatos (título, obra, fecha) cuando están disponibles. Se documentan criterios de inclusión/exclusión y licencias. (2) Distribución por clase: Conjunto prácticamente balanceado: $n_{\text{PS}} = 96$ y $n_{\text{AMOR}} = 84$ ($N = 180$). Véase la Figura 1 para el conteo por categoría. (3) Preparación: Se corrigieron únicamente errores evidentes de codificación/ortografía, preservando versificación, estrofas, puntuación y diacríticos. Además se verificó la unicidad por título y texto; en coincidencias casi idénticas, se conserva una sola versión.

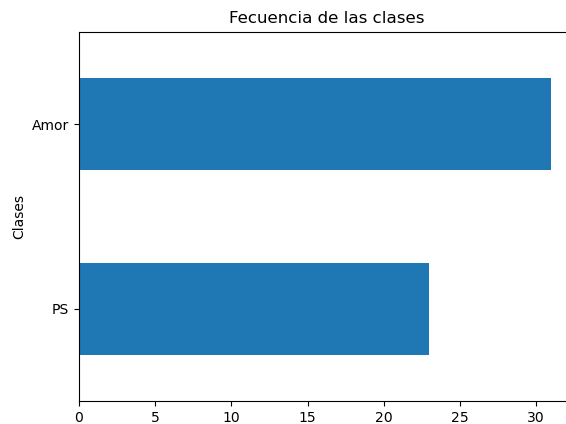


FIGURA 1. Conteo de poemas por categoría. *PS: Político–Social.

Modelos comparados

Para la experimentación se evaluaron cuatro modelos de lenguaje en español: BERT base cased, BERT base uncased, DistilBERT y RoBERTa (BERTIN).¹

- **BERT-base *cased*:** dccuchile/bert-base-spanish-wwm-cased.
- **BERT-base *uncased*:** dccuchile/bert-base-spanish-wwm-uncased.
- **DistilBERT ES *uncased*:** dccuchile/distilbert-base-spanish-uncased.
- **RoBERTa-base (BERTIN):** bertin-project/bertin-roberta-base-spanish.

Estrategias de uso

1. **Fine-tuning:** codificador + cabeza densa (2 salidas), entrenamiento conjunto con entropía cruzada; selección del mejor *checkpoint* por F1 macro en validación.
2. **Feature-based:** codificador congelado, representación fija por poema mediante *mean pooling* del `last_hidden_state` (enmascarando [PAD]), y clasificador lineal (regresión logística).

Preprocesamiento y tokenización

Se empleó tokenización por subpalabras, alineada con el preentrenamiento de cada modelo. A modo de ejemplo, con WordPiece (BERT) la segmentación para dos versos de *Los enemigos* introduce [CLS] al inicio, [SEP] al final y [PAD] para relleno; en RoBERTa se usan <s>, </s> y <pad>. Las variantes

¹DistilBERT es una versión destilada de BERT que conserva alta efectividad con menor costo computacional.

uncased convierten a minúsculas, mientras que las *cased* y RoBERTa preservan mayúsculas y acentos. En WordPiece, el prefijo ## denota continuación de palabra.

Las secuencias se truncaron a `max_length = 512` y se aplicó *padding* dinámico por lote, preservación de saltos de verso/estrofa, puntuación y diacríticos. Se usó `attention_mask` para ignorar [PAD], donde la *attention mask* marcó con 1 los tokens válidos y con 0 el relleno. Se utilizó el tokenizador nativo de cada modelo.

Configuración de entrenamiento

Para realizar la experimentación el dataset se particionó de manera proporcional y reproducible con `seed = 42`. Primero, `[train : test] = [70 % : 30 %]`; luego, el 10 % de train se destinó para validación, resultando `[train : val : test] = [63 % : 7 % : 30 %]`. Por lo tanto con $N = 180$: $N_{\text{train}} = 113$, $N_{\text{val}} = 13$ y $N_{\text{test}} = 54$. La misma partición se reutiliza en todas las comparaciones, preservando las proporciones por clase en cada subconjunto.

Los hiperparámetros utilizados se fijaron de forma común a todos los modelos y se seleccionaron siguiendo las recomendaciones estándar en la literatura para tareas de clasificación con Transformers en corpus de tamaño reducido [4, 20, 21]. Específicamente, se empleó el optimizador AdamW con tasa de aprendizaje de $5e-5$, un valor ampliamente adoptado en ajuste fino monolingüe para español. El tamaño de lote (`batch size = 16`) se eligió tras evaluar 8, 16 y 32 en un subconjunto de validación preliminar, priorizando estabilidad en la convergencia sin sobreajuste. El número máximo de épocas (32) y la paciencia para la detención temprana (4 épocas sin mejora en F1 macro sobre validación) se establecieron para permitir convergencia suficiente incluso en los modelos más lentos, mientras se evitaba sobreentrenamiento.

La misma configuración se aplicó a los cuatro modelos sin ajuste individualizado, garantizando así que las diferencias de desempeño reflejen propiedades intrínsecas de las arquitecturas y no ventajas derivadas de una sintonización selectiva de hiperparámetros.

Métricas

Como métrica primaria se determinó usar F1–macro. Se reportan además exactitud (Acc), precisión macro, recall macro y matrices de confusión normalizadas. Salvo indicación, las métricas se calculan sobre el conjunto de pruebas fijo (70 / 30; `seed = 42`). Cuando se promedian r corridas (misma partición, diferentes inicializaciones), se informa $\text{media} \pm \text{desviación estándar}$.

Selección de modelo y detención temprana

El mejor modelo se elige por F1–macro en validación. Entrenamiento con detención temprana tras 4 épocas sin mejora (límite superior: 32). El mejor *checkpoint* se congela antes de evaluar en el conjunto

de prueba.

Reproducibilidad

La semilla global se estableció en 42 y misma partición estratificada [63 %:7 %:30 %] (train:val:test) reutilizada en todas las comparaciones. En cuanto al entorno de ejecución, se tienen las siguientes especificaciones: Python 3.12.7; SO/Plataforma Linux-6.11.0; PyTorch 2.8.0; Transformers 4.55.0; Datasets 4.0.0; Tokenizers 0.21.4; NumPy 2.0.1; Pandas 2.3.1; CUDA disponible: True; versión CUDA: rocm6.4; cuDNN: 3004000; GPU: AMD Radeon RX 7800 XT (1 dispositivo).

RESULTADOS

Esta sección presenta los hallazgos empíricos del estudio comparativo entre cuatro arquitecturas *Trans-former* en la clasificación temática de poesía en español. Se reportan los resultados obtenidos bajo dos estrategias de adaptación (*feature-based* y *fine-tuning*), evaluadas sobre un conjunto de prueba fijo de 54 poemas. El análisis incluye métricas cuantitativas (accuracy, precisión, recall y F1 macro), matrices de confusión, proyecciones UMAP de las representaciones latentes y un examen cualitativo de los patrones de error. El objetivo es contrastar el desempeño absoluto de cada modelo, y de cómo sus diferencias arquitectónicas y de tokenización se traducen en comportamientos distintos ante las ambigüedades propias del lenguaje poético.

Tratamiento del conjunto de datos

La Figura 2 compara la distribución por categoría (train: 113; val: 13; test: 54). En los tres casos se mantiene la ligera mayoría de *Político-Social*, confirmando que la partición estratificada conservó el balance de forma consistente.

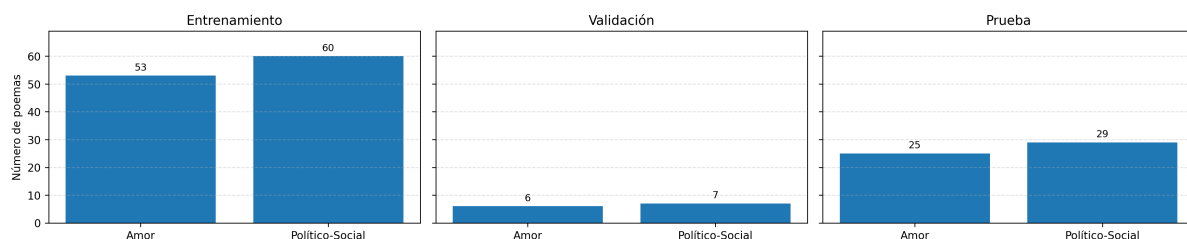


FIGURA 2. Distribución por categoría en los tres subconjuntos: entrenamiento, validación y prueba.

Resultados del Transformer como extractor de características

Luego de que se congela el codificador y se obtienen representaciones fijas por poema mediante *mean pooling* de *last_hidden_state* (enmascarando [PAD]). Sobre dichas representaciones se entrena un clasificador lineal (regresión logística) con la misma partición estratificada. La Tabla 1 resume las métricas macro por modelo; la Figura 3 muestra las matrices de confusión.

Modelo	Accuracy	Prec.-macro	Recall-macro	F1-macro
BERT-base uncased (DCC)	0.926	0.925	0.925	0.925
BERT-base cased (DCC)	0.926	0.939	0.920	0.924
RoBERTa-base (BERTIN)	0.852	0.851	0.851	0.851
DistilBERT uncased (DCC)	0.852	0.854	0.848	0.850

Tabla 1. Resultados con extractor de características (clasificador lineal).

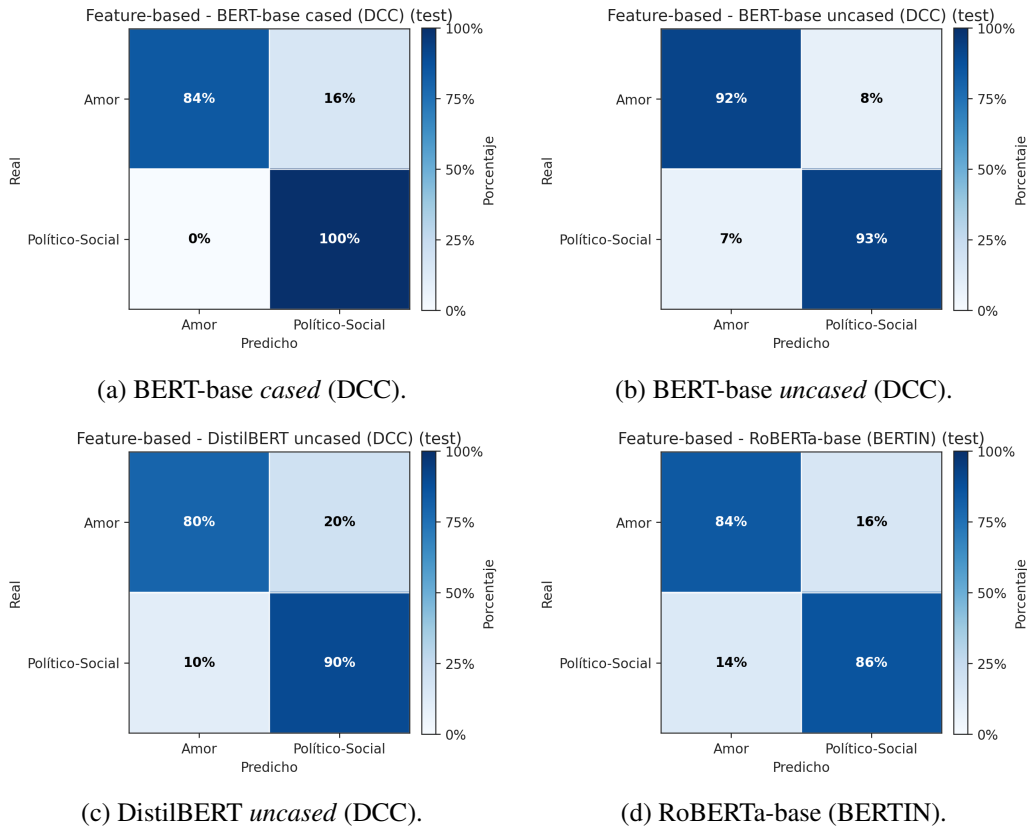


FIGURA 3. Matrices de confusión con enfoque *feature-based*.

Resultados de clasificación

La Tabla 2 resume el desempeño por modelo en *test*, mostrando que RoBERTa-base (BERTIN) alcanza el F1 macro más alto (0.926), seguido por BERT-base *cased* (0.889). Las matrices de confusión de la Figura 4 confirman que RoBERTa comete el menor número de errores y los más equilibrados entre clases.

Modelo	Accuracy	F1 macro	Prec. macro	Rec. macro
RoBERTa-base (BERTIN)	0.926	0.926	0.931	0.931
BERT-base <i>cased</i> (DCC)	0.889	0.889	0.903	0.897
BERT-base uncased (DCC)	0.870	0.870	0.891	0.879
DistilBERT uncased (DCC)	0.870	0.870	0.891	0.879

Tabla 2. Resultados con afinación del modelo (mejor *checkpoint* por validación; métricas en *test*).

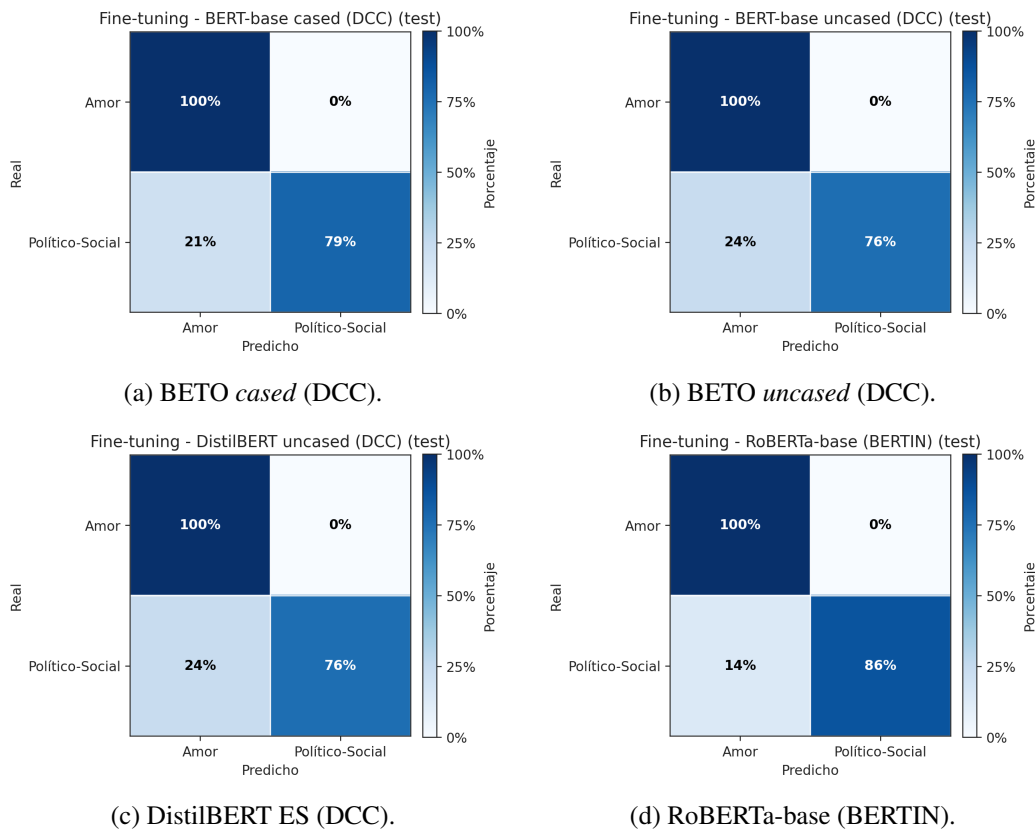


FIGURA 4. Matrices de confusión en *fine-tuning*.

El análisis comparativo de las Tablas 1 y 2 revela dos hallazgos. Primero, la estrategia de fine-tuning invierte el orden de desempeño observado en el enfoque *feature-based*: mientras que en este último los

modelos BERT (DCC) superan a RoBERTa (BERTIN), tras el ajuste fino es precisamente RoBERTa quien alcanza el F1 macro más alto (0.926). Esto sugiere que RoBERTa se beneficia más del ajuste fino en tareas de alta exigencia semántica como la poesía.

En segundo sitio, DistilBERT muestra un desempeño idéntico al de BERT-base *uncased* en el fine-tuning ($F1 = 0.870$), a pesar de tener 40 % menos de parámetros. Esto indica que, en un corpus pequeño, la capacidad expresiva adicional de BERT no se traduce en ganancias, y que DistilBERT conserva suficiente información para la tarea. Sin embargo, su rendimiento en el enfoque *feature-based* es inferior al de BERT.

Estructura latente: proyecciones UMAP

Para inspección cualitativa, se proyectaron a 2D (UMAP, *cosine*, *n_neighbors*=15, *min_dist*=0.1) las representaciones del conjunto de prueba antes (encoder congelado) y después del fine-tuning. En general:

- **Antes:** solapamiento notable entre *Amor* y *Político–Social*; cúmulos incipientes y fronteras difusas, con poca variación entre arquitecturas.
- **Después:** mayor cohesión intra-clase y menor solapamiento visual en todos los modelos, consistente con el aumento de F1 en *test*.

Este patrón sugiere que, en un régimen de datos limitados, el proceso de fine-tuning tiene un impacto más determinante en la organización latente del espacio de representación que las diferencias arquitectónicas iniciales entre los modelos. Es decir, aunque las arquitecturas parten de distintas distribuciones contextuales, el ajuste fino sobre una tarea específica tiende a superar esas diferencias iniciales, alineando las representaciones hacia la separabilidad requerida por la clasificación.

Las proyecciones UMAP (Figura 5) ofrecen una visión complementaria. Antes del *fine-tuning*, todas las arquitecturas muestran un solapamiento significativo entre categorías, con fronteras difusas e inestables. Tras el ajuste fino, en cambio, se observa una clara cohesión intraclassa y separación interclase en los cuatro modelos, lo que visualmente respalda el incremento en F1 macro.

Crucialmente, el cambio inducido por el *fine-tuning* es mucho más pronunciado que las diferencias iniciales entre arquitecturas. Esto sugiere que, en dominios de datos limitados, como la poesía, el protocolo de adaptación tiene un peso mayor que la elección inicial del modelo base.

Análisis comparativo de errores

Aunque todas las configuraciones muestran alta precisión, las matrices de confusión revelan diferencias sutiles en los patrones de error entre arquitecturas:

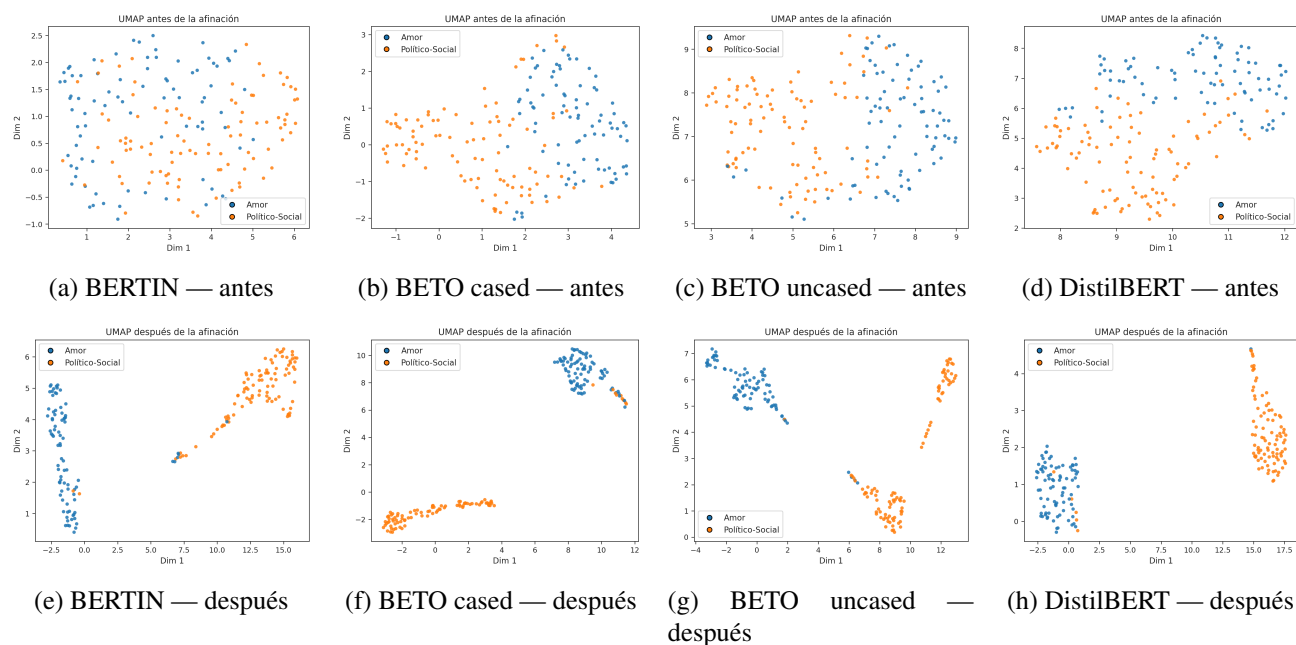


FIGURA 5. UMAP en *test*: cuatro modelos (columnas) con estados *antes* y *después* de la afinación (filas).

- **BERT-base (*cased* y *uncased*)**: cometen principalmente errores de tipo *falso negativo* en la clase *Amor* (es decir, poemas de amor clasificados como *Político-Social*), especialmente en textos con metáforas históricas o referencias colectivas.
- **RoBERTa-base (BERTIN)**: tras el *fine-tuning*, muestra el menor número total de errores (solo 4 en el conjunto de prueba) y los más equilibrados entre clases, lo que se refleja en su F1 macro más alto (0.926).
- **DistilBERT *uncased***: presenta una tendencia ligeramente mayor a *falsos positivos* en *Político-Social*, posiblemente por su menor capacidad para capturar matices contextuales finos dada su arquitectura reducida.
- En general, el enfoque *feature-based* incrementa la confusión en todos los modelos respecto al *fine-tuning*, con errores más concentrados en la clase minoritaria (*Amor*), lo que sugiere que el ajuste fino mejora la sensibilidad a señales temáticas sutiles.

Estos patrones cualitativos concuerdan con las métricas cuantitativas y refuerzan la hipótesis de que la capacidad contextual fina, especialmente en arquitecturas completas como RoBERTa, es clave para distinguir motivos ambiguos en poesía.

El análisis cualitativo de los errores confirma que la principal fuente de confusión radica en poemas con motivos híbridos o metáforas densas. Por ejemplo, textos que vinculan el amor con la lucha social

(“amor colectivo”) o que usan la naturaleza como símbolo político generan ambigüedad razonable incluso para humanos. No obstante, RoBERTa demuestra mayor robustez en estos casos frontera, lo que atribuimos a su capacidad para modelar dependencias contextuales de largo alcance y su sensibilidad a matices léxico-semánticos.

En contraste, las versiones *uncased* (BERT y DistilBERT) pierden información valiosa al normalizar mayúsculas y acentos, lo que en poesía puede eliminar pistas estilísticas críticas (por ejemplo, el uso deliberado de “Patria” en mayúscula para evocar nacionalismo). Esto pone en evidencia la importancia de preservar la forma textual original en dominios literarios.

CONCLUSIONES

Este trabajo evaluó el impacto de la tokenización y de las estrategias de entrenamiento (extractor de características vs. ajuste fino) en modelos *Transformer* en español para la clasificación temática de poesía. Bajo una configuración comparable (misma partición y F1 macro como métrica primaria), encontramos que la tokenización condiciona directamente el desempeño: las diferencias entre WordPiece y byte-level BPE afectan la cobertura de diacríticos y clíticos, la longitud en tokens, el riesgo de truncado y el costo computacional, lo que en conjunto determina el techo de calidad alcanzable. Asimismo, el ajuste fino del *backbone* supera sistemáticamente el uso del modelo como simple extractor de características al reducir la confusión de *Amor* hacia *Político–Social* y lograr un desempeño más equilibrado entre categorías.

En términos de arquitecturas, los modelos tipo RoBERTa en español ofrecieron un buen compromiso entre estabilidad de convergencia y rendimiento, mientras que BERT-ES con *whole-word masking* mostró robustez en segmentaciones difíciles. Por su parte, las variantes destiladas (p.ej., DistilBERT-ES) destacaron por su eficiencia inherente ampliamente documentada en la literatura [7, 21], con una penalización moderada en F1, resultando adecuadas cuando existen restricciones de latencia o recursos. Finalmente, la repro-ducibilidad se confirmó como un factor crítico: fijar semillas, mantener la partición constante y aplicar *early stopping* coherente permitió comparaciones limpias y resultados replicables.

Implicaciones prácticas

Cuando el objetivo es maximizar calidad y se dispone de GPU, conviene preferir el ajuste fino de un modelo RoBERTa-ES o BERT-ES con *whole-word masking*, apoyado por *early stopping* y regularización ligera. Para despliegues con restricciones de latencia o memoria, un modelo destilado resulta competitivo y, si es necesario, puede emplearse como *feature extractor* acoplado a un clasificador ligero. En todos los casos, la higiene de tokenización y un preprocesamiento mínimo alineado con el *checkpoint* del modelo generan mejoras desproporcionadas respecto a su costo.

Limitaciones

Este estudio presenta dos restricciones deliberadas que limitan la generalización de sus hallazgos, pero que fueron impuestas para garantizar un análisis comparativo controlado. En primer lugar, el corpus se compone exclusivamente de poemas de Pablo Neruda (Chile, siglo XX), lo que elimina la variabilidad debida a autoría, época histórica y variación geográfica. En segundo lugar, la tarea se reduce a una clasificación binaria entre los temas *amor* y *político-social*, a pesar de que la poesía en español abarca una gama temática mucho más amplia (p. ej., naturaleza, existencial, religiosa, etc.).

Estas decisiones responden a una estrategia metodológica intencional: al fijar el autor y simplificar la taxonomía temática, se aíslan las variables que podrían confundir la evaluación del impacto relativo de las arquitecturas Transformer y sus protocolos de adaptación. Reconocemos, no obstante, que estos factores (la diversidad autoral y la riqueza temática) son esenciales en escenarios reales de clasificación poética, y que su ausencia limita la extrapolación de los resultados a corpus heterogéneos. Por ello, consideramos fundamental abordar estos aspectos en futuras investigaciones mediante la inclusión de múltiples autores, épocas y regiones, así como la extensión a taxonomías multiclase más representativas.

Trabajo futuro

Futuras extensiones pueden evaluar la robustez ante ruido ortográfico, *code-switching* y variación dialectal, adoptar un enfoque *data-centric* que priorice curación, balanceo y *augmentation* semántico, e incorporar adaptación eficiente en parámetros (adapters/LoRA), *prefix tuning* y precisión mixta para abaratar entrenamiento. También es pertinente profundizar en análisis interpretables a nivel de subpalabras y versos para vincular fallos con patrones lingüísticos, y ampliar el conjunto de métricas con calibración, curvas Precision-Recall por clase y mediciones de costo/latencia por predicción.

AGRADECIMIENTOS

Los autores desean expresar su sincero agradecimiento al Tecnológico Nacional de México y al Instituto Tecnológico de Toluca por el apoyo brindado. El primer autor agradece a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) la beca otorgada para sus estudios de posgrado.

DECLARACIÓN DE CONTRIBUCIÓN DE AUTORES Y COLABORADORES

Julio Cesar Santana Becerril: Conceptualización, Metodología, Software, Análisis formal, Investigación, Redacción (borrador original). **Roberto Alejo Eleuterio:** Supervisión, Metodología, Análisis de resultados, Redacción (revisión y edición). **Sonia Mireya Martínez Gallegos:** Supervisión, Validación metodológica, Revisión crítica del manuscrito. **Federico Del Razo López:** Metodología, Análisis formal, Redacción (revisión y edición). **Everardo Efrén Granda Gutiérrez:** Investigación,

Curación de datos, Revisión. **Grisel Miranda Piña:** Recursos, Asesoría técnica, Revisión.

REFERENCIAS

- [1] G. Marco, J. De La Rosa, J. Gonzalo, S. Ros, and E. Gonzalez-Blanco, “Automated Metric Analysis of Spanish Poetry: Two Complementary Approaches,” *IEEE Access*, vol. 9, pp. 51 734–51 746, 2021.
- [2] A. Perez Pozo, J. De La Rosa, S. Ros, E. Gonzalez-Blanco, L. Hernandez, and M. De Sisto, “A bridge too far for artificial intelligence?: Automatic classification of stanzas in Spanish poetry,” *Journal of the Association for Information Science and Technology*, vol. 73, no. 2, pp. 258–267, Feb. 2022.
- [3] J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” *Journal of Big Data*, vol. 6, no. 1, p. 27, Dec. 2019.
- [4] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, and V. Araujo, “Spanish pre-trained bert model and evaluation data,” in *Proceedings of PML4DC at ICLR*, 2020. [Online]. Available: <http://ceur-ws.org/Vol-2421/>
- [5] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, A. Gonzalez-Agirre, C. Armentano-Oller, C. Rodriguez-Penagos, and M. Villegas, “Maria: Spanish language models,” *arXiv preprint arXiv:2107.07253*, 2021. [Online]. Available: <https://arxiv.org/abs/2107.07253>
- [6] PlanTL-GOB-ES, “PlanTL-GOB-ES/roberta-base-bne – model card,” Hugging Face, 2021, roBERTa entrenado con el corpus de la Biblioteca Nacional de España (BNE). [Online]. Available: <https://huggingface.co/PlanTL-GOB-ES/roberta-base-bne>
- [7] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvallo, and V. Araujo, “Albeto and distilbeto: Lightweight spanish language models,” in *Proceedings of the 13th Language Resources and Evaluation Conference (LREC 2022)*. Marseille, France: European Language Resources Association (ELRA), 2022. [Online]. Available: <https://aclanthology.org/2022.lrec-1.457/>
- [8] T. Ceccon Silveira, “Applying BERT language model to poem classification: A study on data imbalance issues,” Universidade Federal do Rio Grande do Sul (UFRGS), Tech. Rep., 2023. [Online]. Available: <https://lume.ufrgs.br/bitstream/handle/10183/259959/001172384.pdf>

- [9] W. Antoun, F. Baly, and H. Hajj, “Arabert: Transformer-based model for arabic language understanding,” in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4)*. Marseille, France: European Language Resources Association (ELRA), may 2020. [Online]. Available: <https://aclanthology.org/2020.osact-1.2/>
- [10] L. Martin, B. Muller, P. J. Ortiz Suarez, Y. Dupont, L. Romary, E. De La Clergerie, D. Seddah, and B. Sagot, “CamemBERT: a Tasty French Language Model,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, 2020, pp. 7203–7219.
- [11] R. Delmonte and N. Busetto, “Stress Test for Bert and Deep Models: Predicting Words from Italian Poetry,” *International Journal on Natural Language Computing*, vol. 11, no. 6, pp. 15–37, Dec. 2022. [Online]. Available: <https://aircconline.com/ijnlc/V11N6/11622ijnlc02.pdf>
- [12] I. N. Santana, R. S. Oliveira, and E. G. S. Nascimento, “Text Classification of News Using Transformer-based Models for Portuguese,” *Journal of Systemics, Cybernetics and Informatics*, vol. 20, no. 5, pp. 33–59, Oct. 2022. [Online]. Available: <https://www.iiisci.org/DOIJSCI/SA702PC22>
- [13] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [14] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [15] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *International Conference on Learning Representations*, 2013. [Online]. Available: <https://api.semanticscholar.org/CorpusID:5959482>
- [16] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Proceedings of EMNLP*, 2014, pp. 1532–1543. [Online]. Available: <https://aclanthology.org/D14-1162/>
- [17] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching word vectors with subword information,” *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017. [Online]. Available: <https://aclanthology.org/Q17-1010/>

- [18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [19] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, “Recent advances in natural language processing via large pre-trained language models: A survey,” *ACM Comput. Surv.*, vol. 56, no. 2, sep 2023.
- [20] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423/>
- [21] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter,” *CoRR*, vol. abs/1910.01108, 2019. [Online]. Available: <http://arxiv.org/abs/1910.01108>
- [22] J. de la Rosa, E. G. Ponferrada, P. Villegas, P. González de Prado Salas, M. Romero, and M. Grandury, “Bertin: Efficient pre-training of a spanish language model using perplexity sampling,” *arXiv preprint arXiv:2207.06814*, 2022. [Online]. Available: <https://arxiv.org/abs/2207.06814>
- [23] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, “Unsupervised cross-lingual representation learning at scale,” in *Proceedings of ACL*, 2020, pp. 8440–8451. [Online]. Available: <https://aclanthology.org/2020.acl-main.747/>
- [24] M. Schuster and K. Nakajima, “Japanese and korean voice search,” in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5149–5152.
- [25] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of ACL*, 2016, pp. 1715–1725. [Online]. Available: <https://aclanthology.org/P16-1162/>
- [26] T. Kudo and J. Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of EMNLP: System Demonstrations*, 2018, pp. 66–71. [Online]. Available: <https://aclanthology.org/D18-2012/>
- [27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Technical Report*, 2019, gPT-2 tokenizer uses byte-level

BPE. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

- [28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, “Roberta: A robustly optimized bert pretraining approach,” *arXiv preprint arXiv:1907.11692*, 2019. [Online]. Available: <https://arxiv.org/abs/1907.11692>
- [29] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, and J. Pérez, “Spanish Pre-trained BERT Model and Evaluation Data,” 2023, version Number: 1. [Online]. Available: <https://arxiv.org/abs/2308.02976>
- [30] A. Gutiérrez-Fandiño, J. Armengol-Estapé, M. Pàmies, J. Llop-Palao, J. Silveira-Ocampo, C. P. Carrino, C. Armentano-Oller, C. Rodríguez-Penagos, A. Gonzalez-Agirre, and M. Villegas, “MarIA: Spanish Language Models,” *Procesamiento del Lenguaje Natural*, pp. 39–60, 2022.
- [31] J. Cañete, S. Donoso, F. Bravo-Marquez, A. Carvalho, and V. Araujo, “ALBETO and DistilBETO: Lightweight Spanish language models,” in *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, J. Odijk, and S. Piperidis, Eds. Marseille, France: European Language Resources Association, Jun. 2022, pp. 4291–4298. [Online]. Available: <https://aclanthology.org/2022.lrec-1.457/>