Clasificación Multi-etiqueta de Literatura Médica: Implementación de un Modelo Basado en PubMed BERT

Resumen

Se desarrolló un sistema de clasificación multi-etiqueta de literatura médica a partir de títulos y resúmenes, combinando un transformer biomédico pre-entrenado (PubMedBERT) concatenado con una red propia entrenable como cabeza de clasificación. Dado el tamaño limitado del conjunto de datos, se empleó BERT principalmente como extractor de representaciones (con congelamiento parcial de capas), mientras que la cabeza densa se diseñó para mejorar la consistencia entre etiquetas mediante ajustes por co-ocurrencias y mecanismos de recalibración por umbrales.

El pipeline incluyó tokenización compatible con el modelo base, padding dinámico, manejo del desbalance mediante ponderación en la función de pérdida y validación hold-out. El desempeño se evaluó con F1 ponderado como métrica principal, complementado con precisión, recall y matrices de confusión multilabel.

A lo largo del proceso se utilizó de manera intensiva Chat GPT (plan PLUS, modelo GPT-5) como herramienta de apoyo para la investigación conceptual, la generación de código y la documentación técnica, lo que permitió acelerar el desarrollo y mantener un registro estructurado de las decisiones. Se documentan las etapas de exploración, preprocesamiento, diseño, entrenamiento y evaluación.

---

1. Introducción

La clasificación automática de literatura médica constituye un reto fundamental en el campo de la minería de textos biomédicos, debido al volumen creciente de publicaciones y la necesidad de organizar la información de manera eficiente. En este reto se propuso construir un sistema capaz de asignar artículos a uno o varios dominios médicos utilizando como insumo únicamente el título y el resumen.

El objetivo consistió en implementar una solución viable en un plazo restringido, priorizando el uso de modelos pre-entrenados adaptados al dominio biomédico y garantizando un rendimiento adecuado medido mediante el F1 score ponderado.

---

2. Análisis exploratorio y comprensión del problema

El conjunto de datos utilizado estuvo compuesto por aproximadamente 3.565 registros provenientes de fuentes como NCBI y BC5CDR, complementados con ejemplos sintéticos para aumentar la diversidad. Cada registro incluyó tres columnas: ‘title’, ‘abstract’ y ‘group’.

Durante la etapa de exploración inicial se detectó un marcado desbalance de clases: algunas etiquetas concentran gran parte de los documentos, mientras que otras aparecen de manera marginal. La distribución de etiquetas se representó mediante un gráfico de barras, el cual puso en evidencia esta disparidad (Fig. 1). Para mitigar dicho desbalance se exploraron soluciones como la ponderación en la función de pérdida (BCE con weights inversos a la frecuencia de cada clase) y la estratificación en los splits de entrenamiento y validación, buscando una representación más justa de las etiquetas minoritarias.

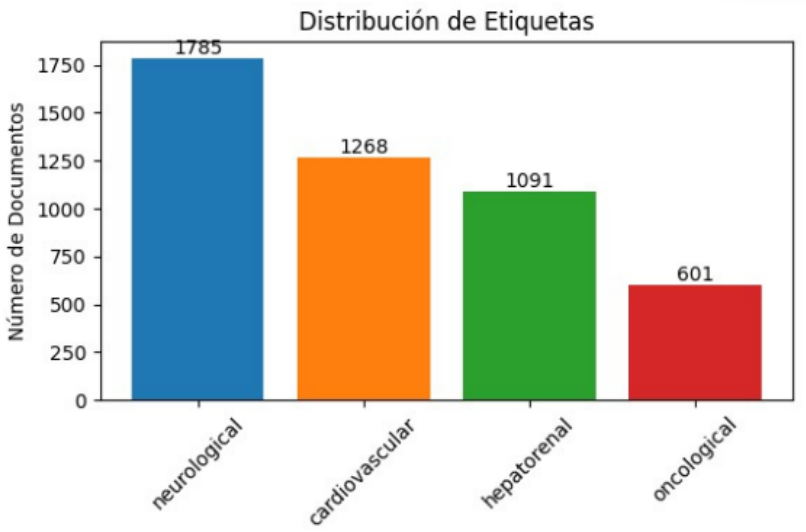

Figura 1. Desbalance del dataset de entrenamiento entregado.

Adicionalmente, se identificó que cada artículo podía estar asociado a una o más etiquetas, lo que confirmó la naturaleza multi-etiqueta y no excluyente del problema. Como ejemplo, la categoría oncológica se presentó en combinación con otros dominios, mostrando un vocabulario muy similar al de áreas relacionadas, lo cual aumentó la dificultad de diferenciación entre clases.

Inicialmente se evaluaron arquitecturas más simples (como redes densas sobre embeddings clásicos), pero la alta similitud entre tópicos biomédicos limitó su desempeño. Esto llevó a la decisión de adoptar un modelo pre-entrenado (PubMedBERT) como extractor de representaciones semánticas, dado que las limitaciones de tiempo y de datos no permitían entrenar un modelo desde cero. Para complementar, se añadió una cabeza entrenable capaz de capturar relaciones adicionales entre etiquetas.

Se implementó una cabeza con atención por etiqueta (Label-Wise Attention, LWA) para resaltar fragmentos relevantes por clase: las salidas token-a-token de PubMedBERT se proyectan a un espacio de atención mediante una capa lineal; cada etiqueta usa su propio vector de consulta para calcular pesos sobre los tokens y realizar un pooling atencional por etiqueta; posteriormente, cada representación pasa por una MLP ligera con dropout que produce los logits por etiqueta, sobre los que se aplica sigmoide y umbrales específicos por clase.

En conjunto, este análisis permitió concluir que la complejidad del problema no radicaba únicamente en la clasificación de términos biomédicos aislados, sino también en la capacidad de modelar interdependencias semánticas entre categorías altamente relacionadas.

---

3. Preparación y preprocesamiento

El pipeline de preprocesamiento contempló varias etapas diseñadas para garantizar la calidad de los datos de entrada y la compatibilidad con el modelo seleccionado.

a. Limpieza:
   Se llevó a cabo una depuración inicial del conjunto de datos que incluyó la eliminación de registros duplicados y el control de valores nulos. Adicionalmente, se aplicaron transformaciones específicas en los textos:
   - Normalización de caracteres como “;”, ya que en algunos casos aparecían dentro del texto sin corresponder a separadores de columnas, lo que afectaba la lectura correcta del archivo CSV.
   - Eliminación de citas numéricas o entre corchetes propias de artículos científicos, que podían introducir ruido irrelevante para el modelo.
   - Remoción de caracteres especiales que no aportaban valor semántico al análisis (por ejemplo, símbolos aislados, secuencias repetidas de guiones).

b. Tokenización:
   Se utilizó el tokenizer oficial de PubMedBERT, asegurando la correspondencia con el espacio semántico del modelo base. El tokenizador permitió descomponer los títulos y resúmenes en subpalabras según el vocabulario biomédico con el que fue entrenado BERT, preservando así información crítica de términos técnicos y acrónimos.

c. Normalización de secuencias:
   Con el fin de mantener la eficiencia y consistencia del entrenamiento, se aplicó padding dinámico y truncamiento de secuencias mediante la clase DataCollatorWithPadding de HuggingFace. Esto garantizó que cada lote de entrenamiento tuviera longitudes uniformes sin desperdicio excesivo de memoria.

d. Manejo del desbalance:
   Dada la fuerte disparidad en la distribución de etiquetas, se incorporó un esquema de

pesos inversos en la función de pérdida Binary Cross-Entropy (BCE), de modo que las clases minoritarias tuvieran un mayor impacto en la optimización. Asimismo, se aplicó una estratificación en la división de los conjuntos de entrenamiento, validación y prueba, favoreciendo que todas las etiquetas, incluso las menos frecuentes, estuvieran representadas de manera proporcional en cada partición.

e. División del dataset:
   La división final se realizó en proporciones de 70 % para entrenamiento, 10 % para validación y 20 % para prueba. Este esquema buscó equilibrar la disponibilidad de datos para la fase de aprendizaje con la necesidad de una evaluación independiente y robusta.

---

4. Selección y diseño de la solución

a. Motivación del enfoque.

Tras probar arquitecturas simples (modelos lineales/MLP sobre embeddings estáticos) y constatar sus límites ante la alta similitud léxica y temática entre dominios biomédicos, se optó por un modelo biomédico pre-entrenado con una cabeza propia. Inicialmente se trabajó con congelación parcial de PubMedBERT para estabilidad y eficiencia; sin embargo, en la versión final se realizó un ajuste completo (fine-tune) del encoder. La cabeza incorpora atención por etiqueta (Label-Wise Attention, LWA) y una MLP ligera, lo que permite resaltar fragmentos específicos por clase y manejar mejor las co-ocurrencias entre dominios. Este enfoque conservó las ventajas del pre-entrenamiento y, al liberar gradualmente todas las capas, mejoró la separación entre clases cercanas sin sacrificar generalización.

b. Arquitectura final implementada.

Texto (title + abstract)

→ Tokenizador PubMedBERT

→ PubMedBERT (ajuste total en B3; congelado en A. Revisar sección 5)

→ Proyección contextual a espacio de atención

→ Atención por etiqueta (Label-Wise)

→ Pooling atencional por etiqueta

→ Logits por etiqueta

→ Sigmoide (probabilidades por etiqueta)

→ Umbrales específicos por etiqueta (optimizados con F1.5)

→ Predicción multilabel

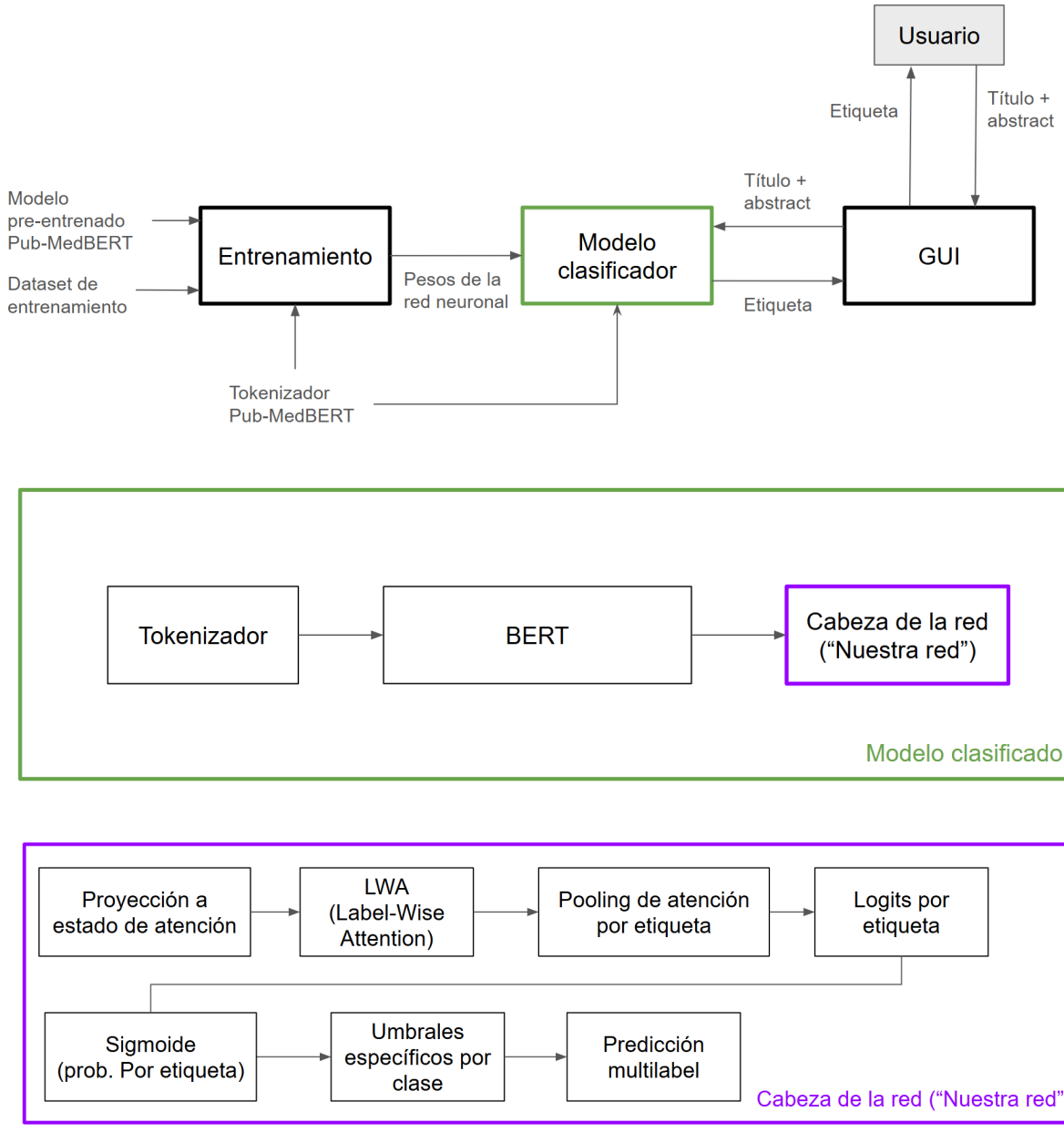

Figure 2. Arquitectura de la solución general (primer diagrama), arquitectura del modelo predictor (segundo diagrama) y arquitectura de la cabeza de la red (tercer diagrama).

c. Adaptación al escenario multietiqueta y desbalance.

- Pérdida: BCEWithLogitsLoss con pos_weight por clase, calculado a partir de frecuencias, para mitigar desbalance.
- Umbrales: selección de umbrales específicos por clase mediante barrido en el conjunto de validación, optimizando el F1 (con énfasis en el F1 ponderado).
- Splits: estratificación en train/val/test para preservar la distribución de etiquetas.

d. Criterios de diseño y decisiones clave.

- Generalización con costo acotado: congelar el encoder y entrenar solo la cabeza reduce parámetros y acelera el ajuste.
- Simplicidad operativa: cabeza ligera con LWA y MLP, evitando componentes no esenciales y facilitando el mantenimiento.
- Reproducibilidad: fijación de seeds, configuración explícita (batch size, max_len, epochs_head/epochs_full) y persistencia de artefactos (labels.json y thresholds.npy) para inferencia consistente.

e. Riesgos y mitigaciones efectivas.

- Sobreajuste: reducción de parámetros entrenables (encoder congelado) y dropout en la cabeza.

En conjunto, el diseño pre-entrenado + cabeza entrenable resultó coherente con las restricciones de datos/tiempo y la naturaleza multi-etiqueta del problema, ofreciendo un balance práctico entre rendimiento, costo y mantenibilidad.

---

5. Entrenamiento y validación

El plan de entrenamiento se implementó en fases:

- Etapa A (encoder congelado). Se entrenó únicamente la cabeza sobre representaciones fijas. El barrido inicial de umbrales produjo valores muy bajos (~0.10), lo que en validación generó recall casi perfecto pero precisión deficiente (activación excesiva de etiquetas). Esta observación fue clave para refinar el barrido de umbrales: se adoptó un grid fino en espacio de probabilidades y F1.5 por clase. Resultado de referencia: macro-F1 ≈ 0.52 (val).

- Etapa B1 (ajuste parcial). Se descongelaron las 2 últimas capas del encoder con tasas de aprendizaje discriminativas (cabeza > encoder). Esto elevó sustancialmente el rendimiento (macro-F1 val ≈ 0.67) y estabilizó los umbrales hacia valores moderados (p. ej., cardio ≈ 0.40, hepato ≈ 0.45), reduciendo falsos positivos —en particular, la sobreactivación de cardiovascular observada al inicio.

- Etapa B2 (más capas). Ampliar el descongelado a 4 capas no aportó mejoras en nuestro régimen (CPU, pocas épocas), con ligera degradación (macro-F1 val ≈ 0.62). Se descartó por coste-beneficio.

- Etapa B3 (ajuste completo). El fine-tuning total del encoder consolidó el desempeño y, combinado con el barrido de umbrales, llevó a métricas muy altas tanto en el conjunto completo como —críticamente— en hold-out. Los umbrales finales reflejaron distintas estrategias por clase (neurological 0.12 para priorizar recall; oncological 0.85 para alta precisión), consistentes con la semántica del corpus.

Para validación externa, se construyó un split hold-out estratificado multilabel. Con el modelo B3 y los umbrales optimizados, se obtuvieron en test:

- macro-F1 = 0.9727, con F1 por clase: cardiovascular 0.980, hepatorenal 0.979, neurological 0.961, oncological 0.971; y perfiles de precisión/recall equilibrados (p. ej., oncological P 0.983 / R 0.959).
  Estos resultados son coherentes con la validación y sugieren una muy buena generalización pese a las co-ocurrencias y el desbalance.

---

7. Conclusiones

Se implementó un clasificador multi-etiqueta basado en PubMedBERT con una cabeza de atención por etiqueta (LWA). El uso del pre-entrenamiento —combinado con un ajuste gradual del encoder— permitió capturar la semántica biomédica y manejar co-ocurrencias entre dominios.

La estrategia de entrenamiento por fases (A, B1 y B3) y la optimización de umbrales específicos por clase redujeron falsos positivos y mejoraron el equilibrio entre precisión y recall. Con el ajuste completo (B3), las métricas en hold-out fueron altas, lo que respalda la capacidad de generalización del sistema.
Las decisiones clave incluyeron: congelación parcial inicial, tasas de aprendizaje discriminativas, BCEWithLogitsLoss con pos_weight para el desbalance y barridos de umbrales por clase (F1.5). Este conjunto logró un balance práctico entre rendimiento, costo y mantenibilidad.