

UNIVERSIDAD POLITÉCNICA DE MADRID



DISEÑO DE UN SISTEMA AVANZADO DE RECOMENDACIÓN MÉDICA MEDIANTE REDES NEURONALES DE GRAFOS Y EL USO DE MODELOS LLM

Máster Universitario en Ingeniería de Telecomunicación

Gema Marín Núñez

2025

TRABAJO DE FIN DE MÁSTER

Título: Diseño de un sistema avanzado de recomendación médica mediante redes neuronales de grafos y el uso de modelos LLM

Autor: Gema Marín Núñez

Tutor: Dr. Alberto Belmonte Hernández

Departamento: Señales, Sistemas y Radiocomunicaciones (SSR)

MIEMBROS DEL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Fecha de lectura: _____

Calificación: _____

UNIVERSIDAD POLITÉCNICA DE MADRID



DISEÑO DE UN SISTEMA AVANZADO DE RECOMENDACIÓN MÉDICA MEDIANTE REDES NEURONALES DE GRAFOS Y EL USO DE MODELOS LLM

Máster Universitario en Ingeniería de Telecomunicación

Gema Marín Núñez

Resumen

El presente proyecto tiene como finalidad el desarrollo de un sistema avanzado de predicción médica basado en la integración de redes neuronales de grafos y modelos de lenguaje de gran tamaño (LLM, por sus siglas en inglés). La solución propuesta busca generar predicciones precisas mediante el uso de una base de datos estructurada en forma de grafo, lo que permitirá identificar tratamientos óptimos para enfermedades específicas. Como fuente de conocimiento se utilizará NeDRex, una plataforma interactiva orientada al descubrimiento biomédico, que facilita la identificación de módulos de enfermedades y la redefinición de fármacos. Esta base de datos recoge relaciones complejas y multiescalares entre genes, proteínas, enfermedades y medicamentos, configurando un entorno idóneo para el entrenamiento de modelos de grafos enriquecidos con información contextual. La arquitectura del sistema combinará representaciones obtenidas mediante técnicas de graph embedding con información semántica extraída de modelos LLM aplicados sobre la base textual asociada. Esta fusión pretende explotar las capacidades de razonamiento simbólico y contextual de los LLM, integrándolas con la representación topológica de las redes biomédicas para fortalecer el proceso de inferencia y predicción. La validación del sistema se llevará a cabo mediante un conjunto de métricas de rendimiento y precisión, con el objetivo de evaluar su capacidad para recomendar tratamientos de forma fiable y superar los modelos predictivos actualmente existentes en el ámbito clínico. Al finalizar el desarrollo, se espera obtener una herramienta capaz de sugerir terapias farmacológicas personalizadas en función de diagnósticos concretos, abriendo la puerta a nuevas líneas de investigación y mejora continua del sistema.

Palabras clave

“LLM”, “NedRex”, Inteligencia Artificial, “Graph SAGE”, “BERT”, «Pythorch», Redes Neuronales de Grafos, “GNN”, Embeddings

Summary

The goal of this project is to develop an advanced medical prediction system that integrates graph neural networks (GNNs) and large language models (LLMs) to generate accurate and detailed predictions based on a graph-structured biomedical database. The system will leverage NeDRex, an interactive biomedical discovery platform designed to identify disease modules and enable drug repurposing. This database captures complex and multiscale relationships between genes, proteins, diseases, and drugs, providing a robust foundation for training graph-based models enriched with contextual information. The proposed architecture will combine graph embeddings with semantic information extracted through LLMs applied to associated textual data. This hybrid approach aims to harness the reasoning and contextualization capabilities of LLMs while preserving the topological structure of the biomedical networks, ultimately enhancing the system's inference and predictive performance. System validation will involve a comprehensive evaluation using precision and performance metrics, assessing the model's ability to reliably recommend pharmacological treatments and to surpass existing predictive methods in the clinical domain. Upon completion, the system is expected to identify effective treatments for specific diseases, offering a powerful tool for personalized medicine and paving the way for future research and continuous improvements.

Keywords

“LLM”, “NedRex”, Artificial Intelligence, “Graph SAGE”, “BERT”, «Pythorch» , Graph Neural Networks, “GNN”, Embeddings

Agradecimientos

Me gustaría dedicar este trabajo a mi familia y amigos, que han estado presentes durante toda mi etapa en la carrera y ahora en el máster, apoyándome siempre y nunca dejando de creer en mi y en todo lo que puedo conseguir. Por último, quiero hacer una mención especial a mi abuelo, que sé que estaría muy orgulloso de mi.

Índice

1	Introducción y objetivos	1
2	Estado del arte	3
2.1	Estado del Arte en Graph Neural Networks (GNN)	3
2.2	Estado del Arte en Large Language Models (LLMs)	13
3	Desarrollo	20
3.1	Arquitectura del proyecto completo	20
3.2	Fases en la creación de un proyecto de DL	21
3.3	Obtención del Dataset - NeDRex	22
3.4	Preprocesado del Dataset - BERT y BioBERT	25
3.5	Creación del modelo Graph Neural Network (GNN)	27
3.5.1	GraphSAGE	29
3.5.2	Implementación del Modelo GraphSAGE	31
3.5.3	Graph Attention Networks (GAT)	33
3.5.4	Implementación del Modelo GAT	34
3.6	Entrenamiento del modelo GNN	35
3.7	Testeo y despliegue del modelo	37

3.8	Implementación en GitHub	38
4	Resultados	39
4.1	Métricas de evaluación para LLM	39
4.1.1	Comparación de resultados de algoritmos LLM empleados	41
4.2	Métricas de evaluación para redes neuronales de grafos	44
4.3	Resultados de los modelo GNN implementados	46
4.3.1	Configuraciones y parámetros de entrenamiento	48
4.3.2	Comparación de resultados con diferentes configuraciones y enfoques . .	53
4.4	Predicción enlace DrugHasIndication	56
4.4.1	GraphSAGE y Binary Cross Entropy	56
4.4.2	GraphSAGE y Margin Ranking Loss	61
4.4.3	GAT y Binary Cross Entropy	66
5	Conclusiones y líneas futuras	71
	Bibliografía	73
	Anexos	77
A	Aspectos éticos, económicos, sociales y ambientales	77
A.1	Introducción	77
A.2	Descripción de impactos relevantes relacionados con el proyecto	78
A.3	Conclusiones	80
B	Presupuesto económico	82

Listado de figuras

1.1	Comparativa entre el enfoque tradicional de descubrimiento de fármacos y el reposicionamiento de fármacos [1]	1
2.1	Arquitectura de una Graph Convolutional Network [2]	4
2.2	Ejemplo de arquitectura de una Message Passing Neural Network [3]	6
2.3	Arquitectura de GraphSAGE [4]	7
2.4	Arquitectura de PinSAGE [5]	8
2.5	Funcionamiento interno de una Graph Attention Network (GAT) [6]	9
2.6	Arquitectura general de una Temporal Graph Network [7]	11
2.7	Arquitectura general de una Graph Transformer Network [8]	13
2.8	Arquitectura de BERT y sus fases principales [9]	14
2.9	Arquitectura de RoBERTa [10]	15
2.10	Arquitectura de ALBERT [11]	16
2.11	Ejemplo de arquitectura de GPT-2 [12]	18
3.1	Flujo del proyecto completo	20
3.2	Fases en la creación de un proyecto de Deep Learning	22
3.3	Estructura del grafo de NedRex completa	23
3.4	Estructura del grafo del proyecto	24

3.5	Proceso de agregación en una Red Neuronal de Grafos [13]	29
3.6	Enfoque transductivo vs inductivo [4]	30
3.7	Pasos en el proceso de GraphSAGE [4]	31
3.8	Arquitectura detallada de la implementación de GraphSAGE	32
3.9	Esquema del mecanismo de atención en GAT. Cada nodo pondera las representaciones de sus vecinos mediante coeficientes de atención α_{ij} y genera una nueva representación combinando la información más relevante[6].	33
3.10	Arquitectura detallada de la implementación de GAT	35
4.1	Visualización de los embeddings generados con BERT mediante UMAP	42
4.2	Visualización de los embeddings generados con BioBERT mediante UMAP . .	43
4.3	Curva de pérdida durante el entrenamiento con BCE y GraphSAGE	50
4.4	Curva de pérdida durante el entrenamiento con MRL y GraphSAGE	51
4.5	Curva de pérdida durante el entrenamiento (Configuración 3)	53

Listado de tablas

3.1	Número de nodos y relaciones en el grafo biomédico.	24
4.1	Comparativa de métricas BERT vs BioBERT	41
4.2	Comparativa de métricas en el conjunto de validación	54
4.3	Comparativa de métricas en el conjunto de test	55
4.4	Top 5 medicamentos para el modelo GraphSAGE con Binary Cross Entropy . .	58
4.5	Top 5 enfermedades para el modelo GraphSAGE con Binary Cross Entropy . .	60
4.6	Top 5 medicamentos para el modelo GraphSAGE con Margin Ranking Loss . .	62
4.7	Top 5 enfermedades para el modelo GraphSAGE con Margin Ranking Loss . .	64
4.8	Top 5 medicamentos para el modelo GAT con Binary Cross Entropy	67
4.9	Top 5 enfermedades para el modelo GAT con Binary Cross Entropy	69

Glosario

GNN - Graph Neural Network

LLM - Large Language Model

GCN - Graph Convolutional Network

GAT - Graph Attention Networks

MPNN - Message Passing Neural Networks

TGN - Temporal Graph Networks

PyKEEN - Python Knowledge Embeddings

BERT - Bidirectional Encoder Representations from Transformers

NLP - Natural language processing

MLM - Masked Language Model

NSP - Next Sentence Prediction

TGN - Temporal Graph Networks

NER - Named Entity Recognition

QA - Question Answering

GPT - Generative Pretrained Transformer

FDA - Food and Drug Administration

PhRMA - Pharmaceutical Research and Manufacturers of America

MLP - Multilayer Perceptron

DL - Deep Learning

BCE - Binary Cross Entropy

MRL - Margin Ranking Loss

AUC - Area Under the ROC Curve

MRR - Mean Reciprocal Rank

NDCG - Normalized Discounted Cumulative Gain

t-SNE - t-distributed Stochastic Neighbor Embedding

UMAP - Uniform Manifold Approximation and Projection

TP - True positive

FP - False positive

TN - True negative

FN - False negative

TPR - True positive rate

FPR - False positive rate

ECA - Enzima Convertidora de Angiotensina

IECA - Inhibidor de la Enzima Convertidora de Angiotensina

EGFR - Epidermal Growth Factor Receptor

GH - Growth hormone

EPO - Erythropoietin

UCP2 - Uncoupling protein 2

GABA - Gamma-aminobutirico acid

Introducción y objetivos

En el ámbito de la medicina, uno de los mayores desafíos que sigue existiendo a día de hoy es el gran reto de descubrir y desarrollar tratamientos efectivos para diversas enfermedades. El camino tradicional para crear nuevos medicamentos es largo, costoso y lleno de incertidumbres, pudiendo durar más de diez años y costar más de mil millones de dólares. Además, es desalentador saber que, según datos de la Pharmaceutical Research and Manufacturers of America (PhRMA) [14], solo un 12 % de los fármacos que pasan a ensayos clínicos logran aprobación de la FDA (Food and Drug Administration) [15]. Esto resalta la urgente necesidad de métodos innovadores que agilicen y mejoren la identificación de nuevos tratamientos.

Una estrategia novedosa es el reposicionamiento de medicamentos [1], que busca nuevos usos para fármacos ya existentes. Esta técnica puede disminuir tanto el tiempo como los costos del desarrollo de medicamentos y brinda una segunda oportunidad a compuestos cuyo desarrollo se ha detenido. En la figura 1.1 se puede observar la diferencia entre el enfoque tradicional de descubrimiento de fármacos y el reposicionamiento de fármacos. No obstante, el reposicionamiento enfrenta desafíos, como la complejidad de las interacciones biológicas y el manejo de grandes volúmenes de datos, lo que exige soluciones sofisticadas para analizar e interpretar estas interacciones y encontrar candidatos adecuados para el reposicionamiento.

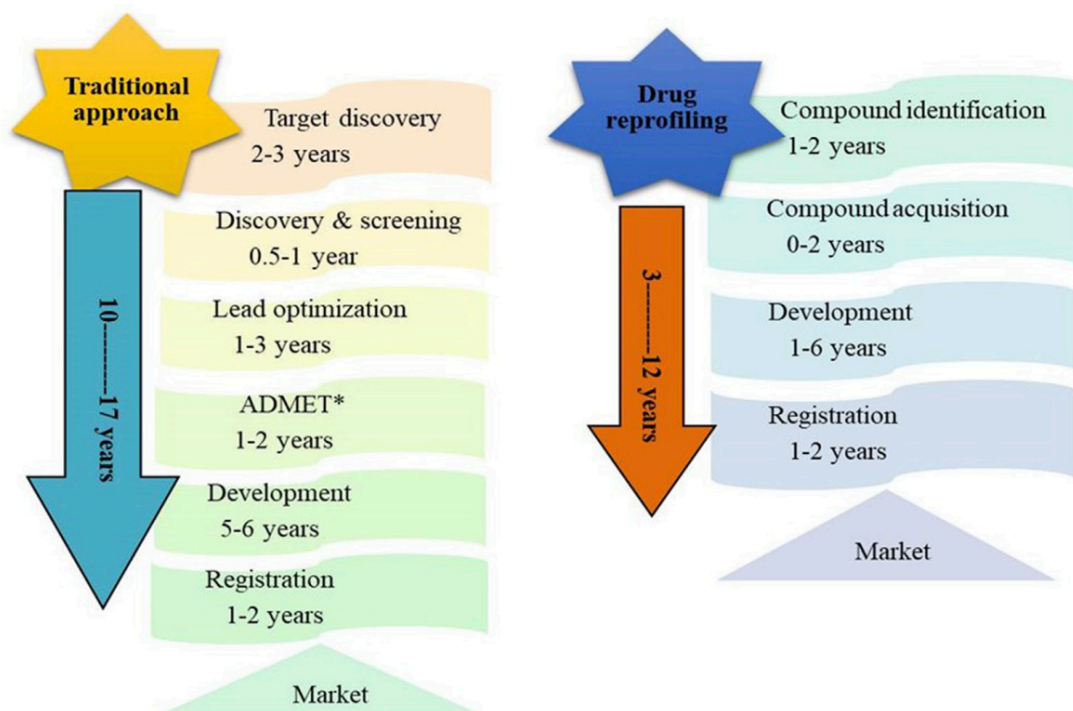


Figura 1.1: Comparativa entre el enfoque tradicional de descubrimiento de fármacos y el reposicionamiento de fármacos [1]

Aquí es donde entra en juego la propuesta de un sistema avanzado de predicción médica que combina dos tecnologías punteras en análisis de datos: redes neuronales de grafos y modelos de lenguaje de gran tamaño (LLM). Este sistema tiene como objetivo abordar la complejidad del reposicionamiento de medicamentos, analizando la extensa red de interacciones entre genes, proteínas, enfermedades y fármacos. Se apoya en NeDRex, una plataforma interactiva que agrupa estas relaciones complejas, para proporcionar predicciones concretas y bien fundamentadas sobre nuevos usos potenciales de medicamentos ya conocidos.

NeDRex se distingue por integrar datos que van desde la biología molecular hasta la farmacología clínica, convirtiéndola en una herramienta esencial para este proyecto. Al aprovechar NeDRex, el sistema no solo accede a un amplio conocimiento científico, sino que también utiliza estructuras de grafos para realizar análisis detallados, descubriendo conexiones no siempre obvias.

La propuesta del trabajo, por tanto, se centra en desarrollar un modelo que, al recibir información específica sobre una enfermedad, pueda sugerir los tratamientos farmacológicos más adecuados y novedosos. Esta iniciativa podría revolucionar el modo en que se descubren medicamentos, mejorando la personalización y eficacia de los tratamientos y avanzando hacia una medicina más focalizada y eficiente.

Implementando este sistema, esperamos aportar significativamente a la medicina, ofreciendo una herramienta que acelere la identificación y el reposicionamiento de fármacos, beneficiando a investigadores, médicos y, especialmente, a los pacientes, quienes obtendrán terapias más rápidas, seguras y efectivas. En definitiva, este proyecto de máster no solo supone un progreso técnico en bioinformática y farmacología, sino que también manifiesta un compromiso con la mejora de la atención médica y el bienestar general de la sociedad.

El trabajo se ha dividido en las siguientes partes:

- Aplicación de modelos LLM para extraer los embeddings de la información de los diferentes nodos de la base de datos de NeDRex para poder entrenar a la red con ellos.
- Aplicación de algoritmos de redes neuronales de grafos para realizar la recomendación automática de posibles candidatos de fármacos asociados a la enfermedad dada.

El trabajo constará por tanto de una sección de **Introducción y objetivos**, tras ello se presentará el **Estado del arte** de algoritmos y soluciones relacionadas con el problema propuesto. La siguiente sección, **Desarrollo**, detallará todos los desarrollos más relevantes llevados a cabo durante el trabajo. Finalmente, una sección de **Resultados** incorporará todas las métricas, resultados numéricos y valoraciones sobre los objetivos buscados. Finalmente, se cerrará el contenido con las **Conclusiones y Líneas Futuras** más relevantes que puedan extraerse.

Estado del arte

En esta sección se describirán las soluciones actuales, dentro del campo de la inteligencia artificial, que pueden ayudar a desarrollar los propósitos del trabajo propuesto, o que se asemejan al problema bajo estudio. En primer lugar se centrará el análisis del Estado del Arte en **Algoritmos de Redes Neuronales de Grafos (GNN)**. La idea es poder usar una red neuronal de grafos para entrenarla con nuestra base de datos de grafos. En segundo lugar se presentarán diferentes técnicas de modelos de aprendizaje automático que pueden comprender y generar un texto en lenguaje humano, **LLM**. Esto lo usaremos para introducir en nuestra red neuronal de grafos nuevas entradas de datos para ayudar a mejorar la precisión de los resultados obtenidos. Finalmente se revisará el estado del arte en soluciones con el mismo propósito que el propuesto en el trabajo.

2.1 Estado del Arte en Graph Neural Networks (GNN)

El aprendizaje profundo ha transformado significativamente diversas áreas del aprendizaje automático en los últimos años, abarcando desde la clasificación de imágenes y el procesamiento de videos hasta el reconocimiento de voz y la comprensión del lenguaje. Tradicionalmente, estos avances han utilizado datos representados en un espacio euclidiano. No obstante, ha crecido el interés en aplicaciones donde los datos provienen de estructuras en forma de grafos, reflejando relaciones complejas y dependencias entre elementos. La naturaleza de los datos en grafos ha presentado retos notables para los algoritmos de aprendizaje automático convencionales. En respuesta, han surgido numerosas investigaciones enfocadas en adaptar y extender las metodologías de aprendizaje profundo para abordar los datos en forma de grafos. Las GNN se pueden clasificar en cuatro tipos distintos: GNN recurrentes, GNN convolucionales, autocodificadores de grafos y GNN espaciotemporales [16]. En este trabajo vamos a usar GNN convolucionales.

Las redes neuronales basadas en grafos se utilizan para analizar y modelar datos estructurados como grafos, donde los nodos representan entidades y los bordes las relaciones entre ellas. Este tipo de redes es fundamental en áreas como la bioinformática, para predecir interacciones proteína-proteína; en las redes sociales, para analizar la estructura y dinámicas de las conexiones entre usuarios; en la investigación sobre el cerebro, para modelar las interacciones neuronales; y en sistemas de recomendación, donde pueden ayudar a prever las preferencias de los usuarios basándose en la estructura de las interacciones previas [17]. A continuación, se van a presentar los algoritmos del estado del arte que han presentado los mejores resultados en los últimos años.

Uno de los modelos más fundamentales en el campo de las GNN es la Graph Convolutional Network (GCN) introducida por Thomas N. Kipf y Max Welling en 2016 [18]. Las redes convolucionales de grafos (GCN) son una clase poderosa de redes neuronales diseñadas para

trabajar con datos estructurados en forma de grafos. La arquitectura GCN combina información de la topología del grafo y las características de los nodos para aprender representaciones de nodos que capturan tanto su contenido como su contexto en el grafo.

La investigación en GCN ha progresado rápidamente para abordar la necesidad de procesar datos complejos en campos como la bioinformática, la detección de fraude y el procesamiento del lenguaje natural. Por ejemplo, las variantes de GCN, como las Graph Attention Networks (GAT), han introducido mecanismos de atención que permiten a la red enfocarse en la información relevante al ponderar la importancia de los nodos vecinos. Esto mejora la flexibilidad y la capacidad de la red. Esta nueva fue presentada por red Petar Veličković y su equipo en 2017 [2].

La arquitectura de una GCN, como se puede ver en la figura 2.1 consta de varias capas de convolución en grafos. Cada capa es como un paso en el que se agregan y transforman los datos. En el primer paso, un nodo recopila información de sus vecinos inmediatos y combina esta información con la suya propia. Esto se hace mediante una operación que pondera la importancia de cada vecino. Después, se aplica una función no lineal que ayuda a la red a aprender representaciones más profundas de los datos. A medida que la información pasa por más capas de la GCN, cada nodo obtiene información de un vecindario más amplio del grafo. Esto permite que la red capture dependencias más complejas en la representación de cada nodo. Finalmente, una capa de salida adapta las representaciones aprendidas a una tarea específica, como clasificar nodos o predecir conexiones entre ellos.

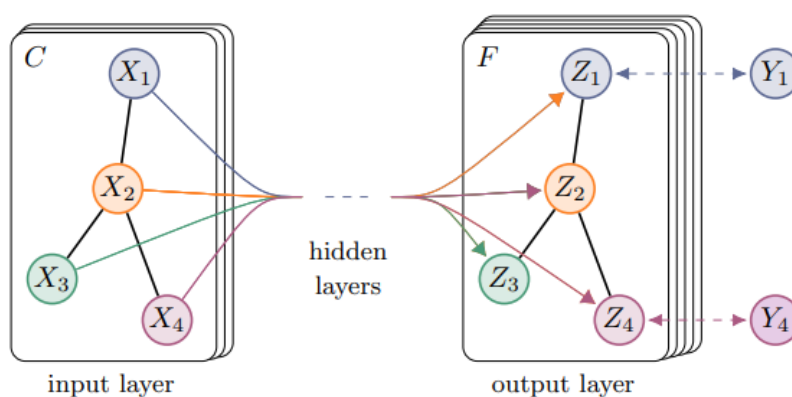


Figura 2.1: Arquitectura de una Graph Convolutional Network [2]

La continua innovación en la arquitectura de las GCN se refleja en trabajos recientes que buscan mejorar la eficiencia del aprendizaje en grafos grandes y abordar desafíos como la interpretación y la resistencia a ejemplos adversos. Estas mejoras abren nuevas posibilidades para aplicar GCN en problemas del mundo real, consolidando su posición como una herramienta fundamental en el aprendizaje automático en grafos.

En 2017 Justin Gilmer introdujo por primera vez Message Passing Neural Networks (MPNN) [3]. Propuso un marco general para el aprendizaje en grafos, enfocándose en la idea de pasar mensajes entre nodos.

Las redes MPNN se basan en que cada nodo en un grafo tiene una representación que se actualiza a medida que recibe y procesa “mensajes” de sus vecinos. Esto es similar a cómo las personas interactúan entre sí y aprenden de su entorno. Por ejemplo, imagina que estás en una fiesta y cada persona representa un nodo en el grafo. Cada persona recibe información de las personas a su alrededor a través de conversaciones y gestos. A medida que reciben estos mensajes, actualizan su conocimiento y se adaptan a la situación. Del mismo modo, en MPNN, cada nodo recibe mensajes de sus vecinos y actualiza su representación en función de estos mensajes. Esto permite que el modelo capture información contextual y relaciones entre nodos en el grafo. En resumen, MPNN es una forma de procesar información en un grafo al actualizar las representaciones de los nodos a través de mensajes de sus vecinos, lo que permite capturar relaciones y contexto en el grafo.

La arquitectura de estas redes, un ejemplo lo podemos ver en la figura 2.2, se basa en capas que se estructuran en una secuencia de fases que facilitan el intercambio de información entre los nodos de un grafo:

- **Capa de Entrada:** Esta capa inicializa las características de los nodos.
- **Capa de Paso de Mensajes:** Durante cada intervalo de tiempo, los nodos emiten y reciben mensajes de y hacia sus nodos vecinos. Los mensajes son transformaciones de las características de los nodos, y pueden integrar también las características de las aristas. Esta capa puede ejecutarse en múltiples iteraciones.
- **Capa de Actualización de Estado:** Tras la recepción de los mensajes, los estados de los nodos se renuevan usando una función de agregación y una función de actualización, como las proporcionadas por GRU o LSTM, para procesar la información recopilada.
- **Capa de Salida:** La información procesada se sintetiza para producir un resultado final, que podría ser la clasificación de nodos, un atributo del grafo en su totalidad o cualquier otro objetivo pertinente. En esencia, las MPNNs manejan la información mediante el intercambio de mensajes entre nodos para discernir la estructura y las relaciones en los datos del grafo.

En 2018 se presenta GraphSAGE [4]. Esta GNN fue propuesta por William L. Hamilton, Rex Ying y Jure Leskovec. Es un método inductivo para el aprendizaje en grafos que permite generalizar a grafos no vistos durante el entrenamiento.

GraphSAGE representa una innovación clave en el campo del aprendizaje de máquina para grafos, destacándose por su habilidad para generar representaciones de nodos de forma

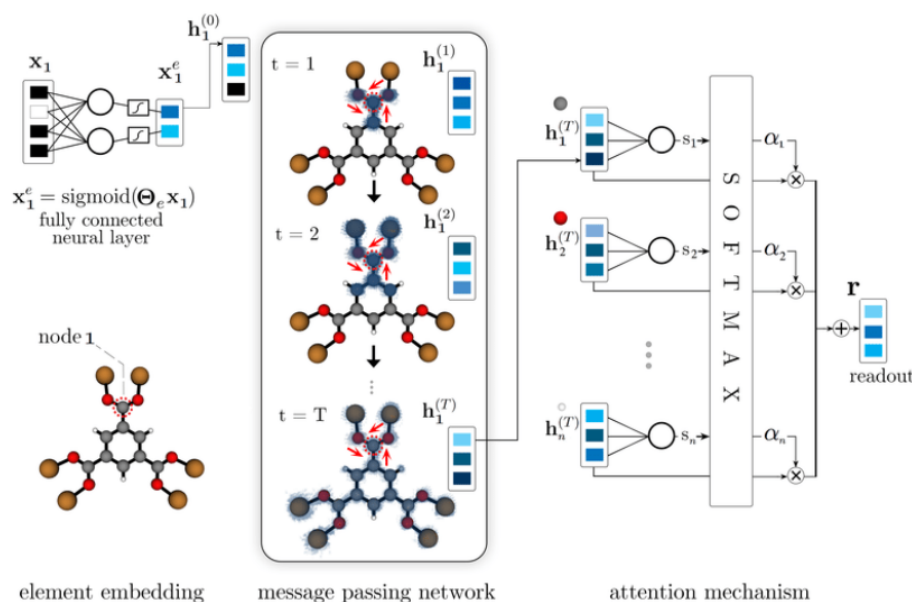


Figura 2.2: Ejemplo de arquitectura de una Message Passing Neural Network [3]

inductiva. A diferencia de métodos transductivos, que necesitan conocer todos los nodos en la fase de entrenamiento, GraphSAGE aprende a generalizar a nodos nuevos mediante una función de agregación. Esta función se encarga de muestrear y combinar características de los vecinos de un nodo a lo largo de varias capas, lo que permite que el modelo se adapte y expanda a nuevos nodos y grafos que surgen, ajustándose a cambios en la estructura del grafo. Esta capacidad inductiva no solo aumenta la eficiencia en el procesamiento, sino que también permite la implementación de modelos en situaciones donde la estructura del grafo es dinámica o no se conoce completamente.

La estructura de GraphSAGE, como podemos ver en figura 2.3 incluye tres etapas fundamentales:

- **Muestreo de Vecindario:** Elige un subconjunto específico de vecinos para cada nodo, lo cual es esencial para manejar grafos de gran escala.
- **Agregación de Características:** Implementa funciones de agregación para combinar las características de los nodos vecinos seleccionados. Estas funciones pueden variar desde promedios simples hasta estructuras más complejas como LSTMs, y se aplican a través de varias capas.^o niveles de vecindad.
- **Predicción y Clasificación:** Utilizando la información característica agregada, la red puede ejecutar tareas como la clasificación de nodos o la predicción de enlaces, considerando tanto la estructura del grafo como las características de los nodos.

GraphSAGE está diseñado para crear representaciones vectoriales (embeddings) que capturan tanto las propiedades individuales de los nodos como sus contextos dentro del grafo más amplio.

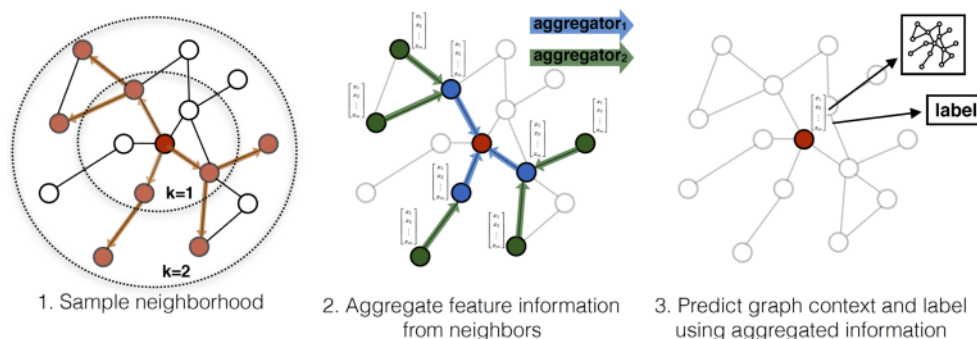


Figura 2.3: Arquitectura de GraphSAGE [4]

Como ampliación a GraphSAGE surge PinSAGE desarrollada por Pinterest [5]. PinSAGE combina métodos de convolución en grafos con estrategias de muestreo aleatorio para manejar de manera efectiva el gran volumen y la naturaleza cambiante de los datos en Pinterest. Emplea el aprendizaje inductivo para ofrecer recomendaciones personalizadas, examinando las conexiones entre los pines y los intereses de los usuarios, aprovechando así las complejas relaciones en el grafo de interacción del usuario. Su desarrollo muestra un esfuerzo considerable en la implementación de aprendizaje automático en grafos en un ambiente productivo real y a la escala de internet.

PinSAGE mejora significativamente GraphSAGE, enfocándose en la eficiencia computacional y la precisión de las recomendaciones en extensos conjuntos de datos. Aunque GraphSAGE fue creado para extender la creación de embeddings de nodos a grafos desconocidos, PinSAGE refina este método para ajustarse mejor a la magnitud y particularidades de los sistemas de recomendación. Incorpora una técnica de muestreo selectivo de nodos clave y emplea convoluciones para formar representaciones de nodos que reflejan la estructura local y las propiedades de los nodos, facilitando así la producción de recomendaciones personalizadas de forma efectiva.

La arquitectura de PinSAGE ilustrada en la figura 2.4 demuestra una metodología de aprendizaje basada en grafos para mejorar los sistemas de recomendación. En esencia, PinSAGE emplea un procedimiento bifásico para crear representaciones de nodos, conocidos como pines en el contexto de Pinterest:

- **Convolución:** Este proceso inicia con un grafo base y aplica secuencialmente una serie de convoluciones para asimilar la información de las características de los nodos adya-

centes al nodo de interés. Esto se realiza mediante funciones de agregación que evalúan y ponderan la relevancia de cada nodo vecino, renovando así las características del nodo focal. Al repetir la convolución en diversas capas, se facilita la transmisión de la influencia de los nodos adyacentes a través del grafo, enriqueciendo la representación del nodo objetivo.

- **Muestreo y Procesamiento en Bloques:** Para manejar grafos de gran tamaño de manera eficaz, PinSAGE adopta una estrategia de muestreo, seleccionando subgrafos centrados en distintos nodos objetivo en lugar del grafo completo. Al procesar estos subgrafos como un conjunto en paralelo, se logra no solo una mayor escalabilidad sino también una optimización del rendimiento computacional del algoritmo.

Al final del proceso, la red es capaz de generar sugerencias, utilizando las representaciones nodales aprendidas para anticipar cómo un usuario podría interactuar con contenido nuevo, basándose en la afinidad con los nodos previamente explorados.

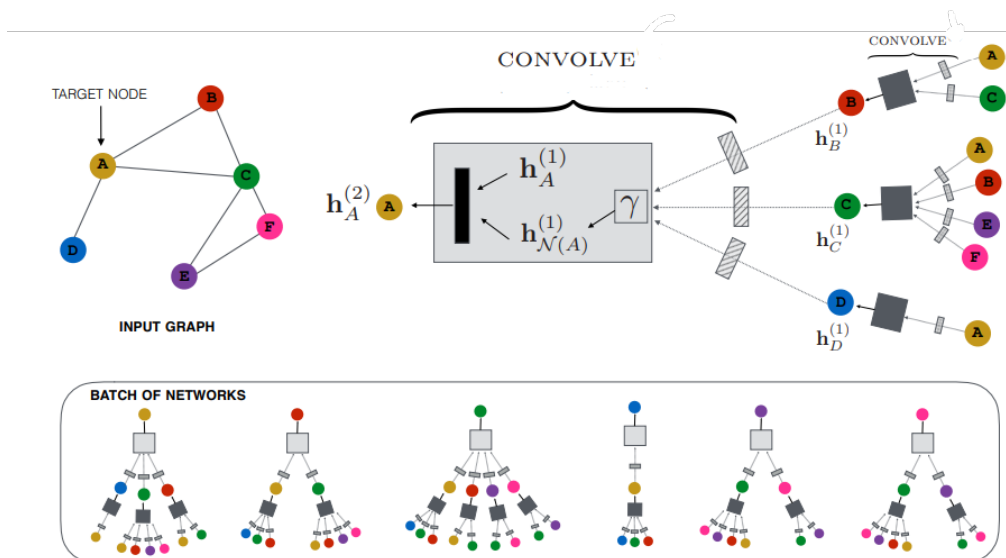


Figura 2.4: Arquitectura de PinSAGE [5]

Petar Veličković y su equipo introdujeron las Graph Attention Network (GAT) [6], una de las extensiones más destacadas de las GNN. Estas redes representan un avance significativo dentro del campo de las redes convolucionales de grafos, al incorporar un mecanismo de atención autoaprendida que permite asignar diferentes pesos a los vecinos de cada nodo durante la agregación.

A diferencia de modelos anteriores de GCN, donde todos los nodos vecinos contribuyen con la misma importancia (normalizada por la estructura del grafo), GAT permite que la red

aprenda qué vecinos son más relevantes para cada nodo. Este mecanismo de atención se implementa mediante un pequeño perceptrón que, para cada par de nodos conectados, estima un coeficiente de atención que indica cuánta influencia debe tener uno sobre el otro. Así, durante la agregación, los mensajes provenientes de los vecinos se ponderan de manera diferenciada, permitiendo una representación más rica y adaptativa.

Este enfoque resulta especialmente útil en contextos donde los nodos pueden tener una conectividad heterogénea o ruidosa, como en aplicaciones biomédicas o sociales, ya que la red puede filtrar o enfatizar información de manera dinámica. Además, al no requerir conocer la estructura global del grafo o su normalización explícita, GAT ofrece una mayor flexibilidad y capacidad de generalización en grafos complejos.

La arquitectura de GAT consta, al igual que otras GNNs convolucionales, de múltiples capas de atención, donde en cada una los nodos actualizan sus embeddings en función de sus vecinos ponderados por atención. Esta estructura permite que, capa a capa, cada nodo incorpore información progresivamente más contextualizada, logrando representaciones altamente expresivas.

En la figura 2.5 se puede ver en detalle su arquitectura, ilustrándose el mecanismo de atención que las caracteriza. En este ejemplo, el nodo 1 (el azul) desea actualizar su representación a partir de la información de sus vecinos conectados, que en este caso serían los nodos 2, 3 y 4. Cada nodo i posee inicialmente un vector de características $\mathbf{h}_i$, que se representa con diferentes colores en la imagen.

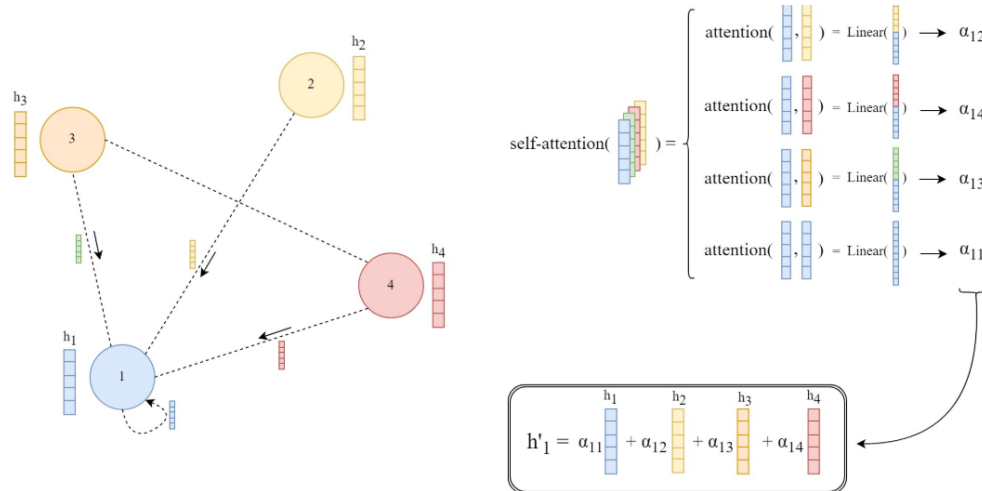


Figura 2.5: Funcionamiento interno de una Graph Attention Network (GAT) [6]

Para combinar esta información, el nodo aplica un mecanismo de *self-attention*: se concatenan los vectores del nodo central y cada uno de sus vecinos, y se pasa esta concatenación por una función lineal seguida de una no linealidad. Esto produce un coeficiente de atención α_{ij} que mide la relevancia del nodo j para el nodo i .

Una vez calculados estos coeficientes de atención, se normalizan (generalmente mediante *softmax*), y se realiza una combinación ponderada de las representaciones vecinas según estos valores. De este modo, la nueva representación del nodo 1, $\mathbf{h}'_1$, se construye como una suma ponderada de sus vecinos, de la siguiente forma:

$$\mathbf{h}'_1 = \alpha_{11} \cdot \mathbf{h}_1 + \alpha_{12} \cdot \mathbf{h}_2 + \alpha_{13} \cdot \mathbf{h}_3 + \alpha_{14} \cdot \mathbf{h}_4$$

Este mecanismo se aplica simultáneamente en todos los nodos del grafo. Esto permite que el modelo se enfoque dinámicamente en las relaciones más informativas para cada nodo, ofreciendo una mejor adaptabilidad frente a ruido estructural o vecinos poco relevantes, lo que lo hace particularmente eficaz en dominios heterogéneos y complejos.

Las Redes de Grafos Temporales (Temporal Graph Networks, TGN) [7] son una de las innovaciones más recientes y destacadas en el campo de las redes neuronales para grafos hasta la fecha. Estas redes van más allá de analizar simplemente la estructura y las características estáticas de los nodos. Se adentran en cómo estos nodos y sus interacciones cambian y se desarrollan a lo largo del tiempo. Esta capacidad es especialmente valiosa en entornos donde las conexiones y los estados son fluidos, como en las redes sociales o en las interacciones biológicas. Lo que hace únicas a las TGN es su componente de memoria, que refresca la representación de cada nodo tras cada evento nuevo, permitiendo así que la red acumule y utilice un historial de interacciones temporales. Este enfoque mejora notablemente la capacidad de la red para hacer predicciones en entornos de grafos que están en constante cambio.

En la figura 2.6 se puede observar la arquitectura de una Red de Grafos Temporales (TGN). En la que destacan los siguientes componentes:

- **Node Embeddings:** Los nodos se representan como vectores de características, conocidos como embeddings, que resumen sus propiedades esenciales. Estos embeddings varían en función del tiempo, ejemplificado por $z_1(t_1)$, $z_2(t_1)$, $z_2(t_2)$, $z_3(t_2)$ donde t_1 y t_2 denotan instantes de tiempo distintos.
- **Decodificador (Decoder):** El decodificador evalúa los embeddings de los nodos para calcular la probabilidad de que existan conexiones entre ellos en diferentes tiempos. Por ejemplo, $p((1, 2) | t_1)$ representa la probabilidad de un enlace entre los nodos 1 y 2 en el instante t_1 .
- **Función de Pérdida (Loss Function):** Se emplea una función de pérdida para medir la diferencia entre las conexiones pronosticadas y las reales observadas en los datos. El ajuste de los parámetros del modelo busca minimizar esta discrepancia.
- **Mensajes:** Los nodos generan mensajes basados en sus interacciones, como $m_1(t_1)$ del nodo 1 al nodo 2, o $m_2(t_1)$ y $m_2(t_2)$ del nodo 2 al nodo 3 en momentos diferentes. Estos mensajes reflejan la información transmitida entre nodos.

- **Mensajes Agregados:** Los mensajes enviados simultáneamente se combinan para consolidar la información de múltiples interacciones en un único vector de características, optimizando el flujo de información.
- **Memoria Actualizada (Updated Memory):** La memoria de cada nodo, que evoluciona a lo largo del tiempo, se actualiza con base en los mensajes agregados, como se muestra con $s_1(t_1)$, $s_2(t_2)$ y $s_3(t_2)$ reflejando el estado de memoria más reciente de los nodos en los respectivos instantes.

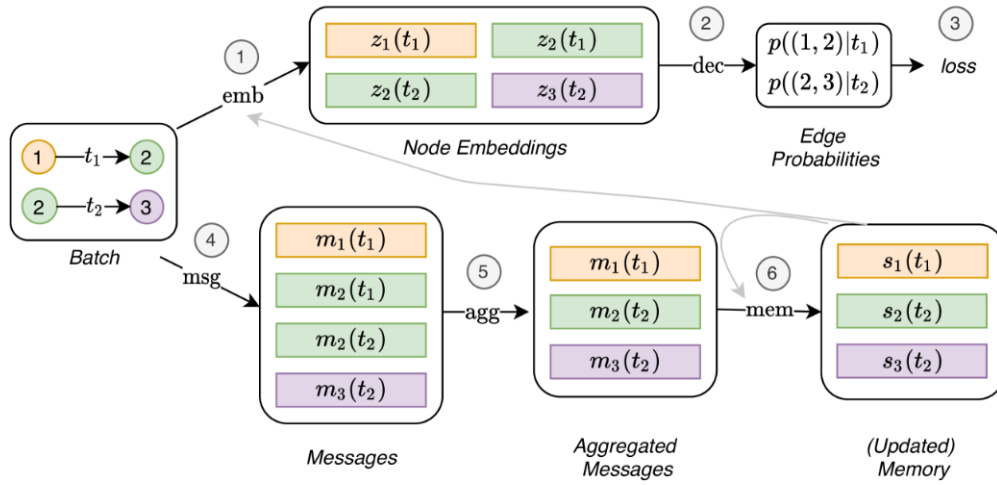


Figura 2.6: Arquitectura general de una Temporal Graph Network [7]

Este ciclo de procesamiento en una TGN permite que la red capture dinámicas temporales y evoluciones en la estructura del grafo, siendo fundamental para aplicaciones como la predicción de enlaces y la clasificación de nodos en grafos que experimentan cambios a lo largo del tiempo.

Además, es relevante destacar la existencia de bibliotecas que facilitan la implementación y experimentación con este tipo de modelos. Una de estas bibliotecas es PyKEEN (Python KnowlEdge EmbeddiNGs) [19].

PyKEEN (Python Knowledge Embeddings) es una librería de Python [20] dedicada a la creación, comparación y evaluación de modelos de embeddings para grafos. Esta biblioteca proporciona un marco de trabajo relevante en el ámbito más amplio de las GNN. Algunas de las características más importantes de PyKEEN son las siguientes:

- **Implementaciones Variadas:** Ofrece implementaciones de múltiples modelos de embeddings de conocimiento, como TransE, DistMult, ComplEx y ConvE. Estos modelos son útiles para aprender representaciones densas y de baja dimensión de los nodos y las aristas en un grafo.

- **Facilidad de Uso:** PyKEEN está diseñado para ser intuitivo, permitiendo a los usuarios configurar experimentos a través de archivos de configuración o directamente en código Python. Esto hace que sea accesible tanto para principiantes como para expertos en el campo.
- **Amplias Capacidades de Evaluación:** La librería provee herramientas robustas para la evaluación de los modelos, permitiendo a los usuarios medir el rendimiento de sus embeddings utilizando una variedad de métricas y protocolos de evaluación.
- **Interoperabilidad:** PyKEEN permite la interacción con otras librerías de Python populares en el ecosistema de ciencia de datos, como NumPy, SciPy y PyTorch, lo cual facilita la integración de los embeddings en pipelines de aprendizaje automático más amplios.
- **Comunidad Activa:** PyKEEN se beneficia de una comunidad de desarrolladores y usuarios activos, y está en constante desarrollo para incluir las últimas innovaciones en el campo de los embeddings de grafos de conocimiento.

En resumen, PyKEEN destaca en su contribución a la comunidad de aprendizaje de representaciones de grafos, al proporcionar una plataforma que agiliza el desarrollo de modelos de embeddings, un componente clave en muchas aplicaciones de las GNN. Esto refuerza el entendimiento de que el avance en el campo no sólo viene por innovaciones teóricas y algoritmos, sino también por el desarrollo de herramientas que hagan esos avances accesibles y prácticos para la investigación y la industria.

En la actualidad, las Redes Neuronales de Grafos (GNN) siguen avanzando, lo que ha llevado a la creación de nuevos enfoques para enfrentarse a los nuevos retos que surgen día a día. Una de las innovaciones más notables es la incorporación de arquitecturas tipo Transformer en las GNN, que se conocen como Graph Transformers [8].

Las GTNs se pueden ver como una evolución de las Redes Neuronales Convolucionales de Grafos (GCNs). A diferencia de las GCNs tradicionales, que realizan operaciones de convolución en estructuras de grafo fijas y homogéneas, las GTNs llevan las cosas un paso más allá. Aprenden a crear nuevas estructuras de grafo que destacan las conexiones importantes entre nodos, lo que les brinda una mayor flexibilidad y capacidad de modelado para aplicaciones más complejas.

Estas redes suponen un importante salto adelante en el mundo de las Redes Neuronales de Grafos (GNN) incorporando mecanismos de atención de Transformadores para procesar eficazmente datos estructurados en grafos. Esta innovadora arquitectura permite a las GNN capturar relaciones complejas y dependencias de largo alcance entre nodos, superando las limitaciones de los enfoques tradicionales de paso de mensajes.

La arquitectura de los GTNs se basa en la creación de nuevas estructuras de grafo a través de capas de transformación de grafos (GT layers). Estas capas tienen la capacidad de detectar

conexiones valiosas entre nodos que, a primera vista, pueden no estar directamente enlazados en el grafo original. Esto permite una representación más rica y efectiva de las relaciones entre los nodos. En la figura 2.7, se muestra cómo las GT layers generan matrices de adyacencia de meta-caminos (meta-path adjacency matrices). Sobre estas matrices, se aplican convoluciones de grafo, similares a las que se utilizan en las GCNs, integrando diversas representaciones de nodos para potenciar el rendimiento en tareas como la clasificación de nodos.

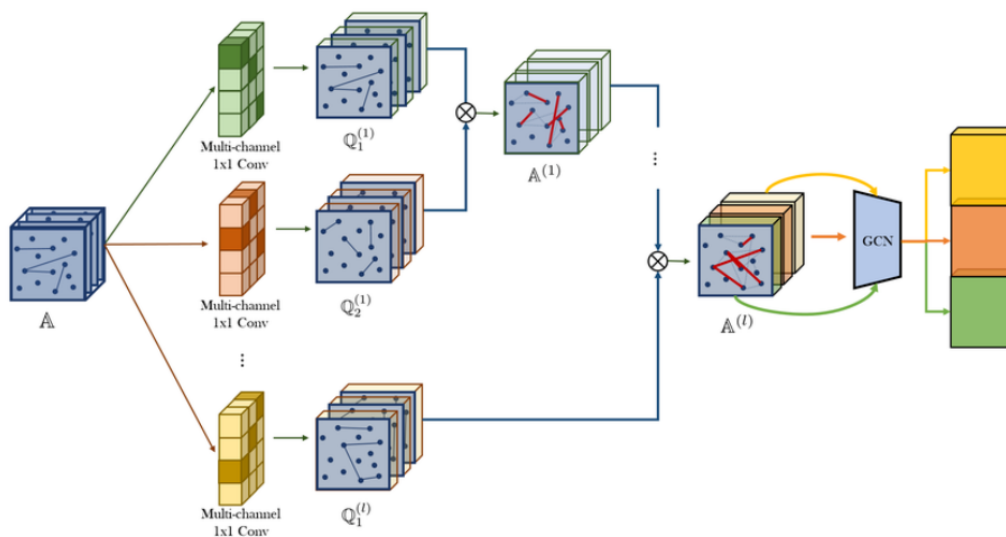


Figura 2.7: Arquitectura general de una Graph Transformer Network [8]

Estas innovaciones son un claro ejemplo del esfuerzo constante que se realiza en el área de la investigación para desarrollar nuevos algoritmos y propuestas para así potenciar las GNN en la modelización de datos complejos y estructurados. Esto está abriendo un mundo de nuevas oportunidades en áreas como la bioinformática, las redes sociales y los sistemas de recomendación.

2.2 Estado del Arte en Large Language Models (LLMs)

En los últimos años, se han creado diversos modelos de lenguaje basados en aprendizaje profundo (LLM), marcando un avance significativo en la capacidad de las máquinas para entender y producir lenguaje natural.

En 2018 Google AI desarrolló BERT (Bidirectional Encoder Representations from Transformers) [9]. BERT ha sido uno de los modelos más influyentes en el procesamiento del lenguaje natural (NLP) por su capacidad para entender el contexto de una palabra en una oración de manera bidireccional. BERT se entrena primero con un extenso conjunto de textos y después

se personaliza para tareas concretas, logrando una mejora notable en diversas actividades de procesamiento del lenguaje natural, como responder preguntas y entender inferencias. Su habilidad para entender contextos complicados en los textos ha marcado un nuevo referente en la interpretación del lenguaje.

En la figura 2.8 se puede observar la arquitectura de BERT con sus dos fases principales:

- **Etapas de pre-entrenamiento:** BERT se somete a un entrenamiento preliminar con textos no etiquetados a través de dos métodos: el Modelado de Lenguaje Enmascarado (MLM) y la Predicción de la Siguiente Oración (NSP). En el MLM, se ocultan aleatoriamente algunas palabras del texto, y BERT aprende a predecirlas observando el contexto que las rodea. Con la NSP, el modelo se entrena para averiguar si una oración B sigue lógicamente a una oración A, ayudando así a BERT a comprender cómo se relacionan entre sí las oraciones.
- **Etapas de Fine-Tuning:** Después de su fase de preentrenamiento, BERT puede ser específicamente afinado para realizar tareas concretas, como la clasificación de textos, el reconocimiento de entidades nombradas o sistemas de preguntas y respuestas. Durante este ajuste fino, se añade una capa de salida adaptada a la tarea en cuestión a la estructura ya entrenada de BERT, optimizando todos los parámetros en función de un conjunto de datos específico y etiquetado.

En ambas etapas, BERT trabaja con una representación de entrada que combina tres tipos de embeddings: de tokens, de segmentos y de posición. Utiliza el token especial [CLS] al comienzo para tareas de clasificación, y los tokens [SEP] sirven para separar pares de oraciones o indicar el final de una.

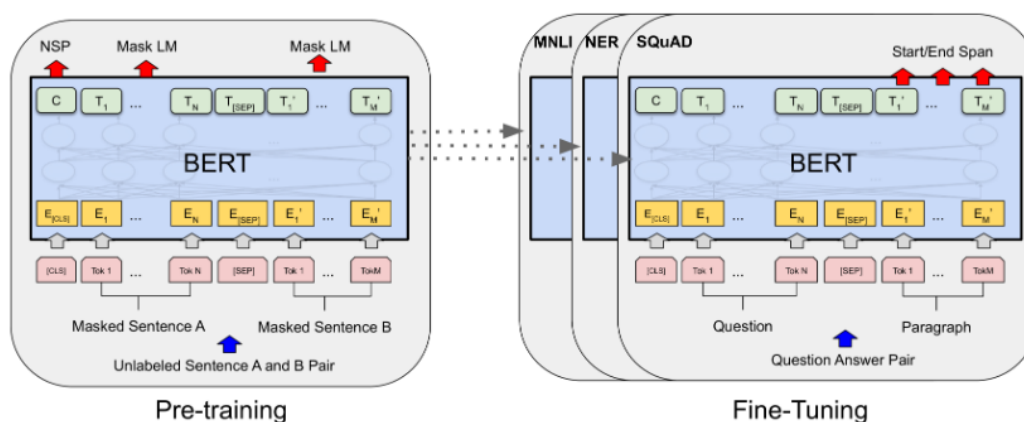


Figura 2.8: Arquitectura de BERT y sus fases principales [9]

Uno de los desarrollos más relevantes basados en BERT para el ámbito biomédico es BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical Text Mining) [21]. Presentado por Lee et al. en 2020, BioBERT se entrenó a partir del modelo BERT original pero utilizando grandes datos biomédicos como PubMed y PMC, lo que lo convierte en una herramienta más adecuada para tareas de procesamiento de lenguaje natural en el ámbito de la biomedicina.

BioBERT ha demostrado un rendimiento superior en tareas biomédicas especializadas como Named Entity Recognition (NER), Question Answering (QA) y relación entre entidades. Al beneficiarse del entrenamiento en un dominio específico, puede captar terminología médica con mayor precisión que los modelos entrenados en lenguaje general. Esta especialización lo ha convertido en una de las principales elecciones para aplicaciones en ciencias de la salud y bioinformática.

A partir de BERT, hay que mencionar que han ido surgiendo nuevos desarrollos. Destaca RoBERTa (A Robustly Optimized BERT Approach) [10] que surgió en 2019.

RoBERTa, una versión avanzada del enfoque BERT, representa un avance significativo en el procesamiento del lenguaje natural. Esta mejora se logra a través de diversas optimizaciones cruciales. En concreto, RoBERTa se beneficia de un entrenamiento más extenso, utilizando lotes de mayor tamaño y una cantidad de datos superior en comparación con BERT. Este enfoque intensivo facilita que RoBERTa capte con mayor precisión las complejidades del lenguaje. Adicionalmente, introduce cambios en la metodología de pre-entrenamiento de BERT, como se puede observar en la figura 2.9, destacando la eliminación de la tarea de Predicción de Siguiente Oración (NSP) y la implementación de ajustes en el enmascaramiento dinámico de los datos. Gracias a estas mejoras, RoBERTa ha logrado igualar o incluso superar a BERT en diversas tareas relacionadas con la comprensión del lenguaje natural.

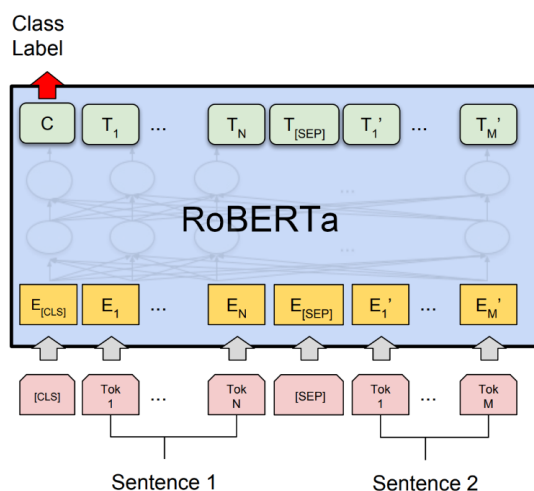


Figura 2.9: Arquitectura de RoBERTa [10]

En 2019, un equipo de expertos de la Universidad Carnegie Mellon y el Grupo de Inteligencia Artificial de Google presentaron XLNet [22]. Este modelo también se construyó sobre los cimientos de BERT, llevando la tecnología un paso adelante. Lo que hace especial a XLNet es su capacidad para preentrenar modelos de lenguaje de manera autoregresiva, empleando una técnica llamada ordenamiento de permutación. Este método le permite entender el contexto de las palabras de manera bidireccional sin necesidad de ocultar partes del texto, algo que BERT sí requería, marcando así una mejora significativa. Gracias a estas innovaciones, XLNet ha mostrado una eficacia sobresaliente en diversas áreas del procesamiento del lenguaje natural.

Otro modelo basado en BERT y mejorando alguna de sus funcionalidades es ALBERT [11], conocido también como "A Lite BERT", es una versión más ligera y optimizada de BERT. Su principal objetivo es mejorar la eficiencia de BERT al reducir la cantidad de parámetros necesarios, sin perder la habilidad de entender y procesar el lenguaje de forma efectiva. Esto se consigue a través de una técnica especial de organización de los componentes del modelo y la reutilización de ciertos elementos en diferentes partes de este. Gracias a estos ajustes, ALBERT no solo mantiene, sino que en algunos casos, supera el rendimiento de BERT en diversas tareas de procesamiento de lenguaje, todo esto mientras usa menos recursos y acelerando el tiempo de entrenamiento.

Su arquitectura se muestra en la figura 2.10, en la que se destaca que las principales diferencias con BERT están en:

- **Factorización de los embeddings de palabras:** ALBERT divide el gran embedding de vocabulario en dos matrices más pequeñas, lo que hace que el modelo sea más compacto sin perder efectividad.
- **Compartir parámetros entre capas:** BERT usa diferentes ajustes para cada nivel de su estructura, mientras que ALBERT usa el mismo ajuste en todos los niveles, lo que reduce su tamaño y hace que el entrenamiento sea más eficiente.

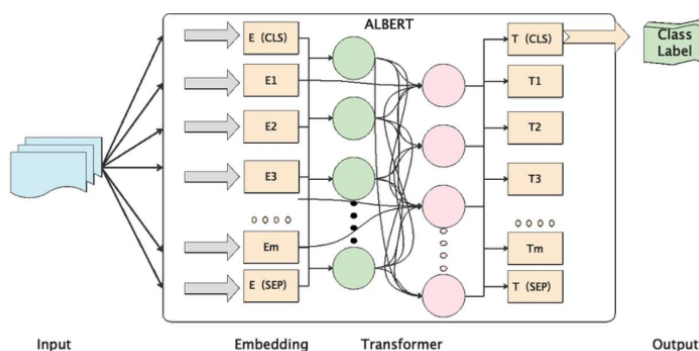


Figura 2.10: Arquitectura de ALBERT [11]

Pero no sólo existen esta familia de LLM. Otro de los modelos más importantes en este campo son los modelos GPT (Generative Pretrained Transformer), desarrollados por OpenAI que se centran en la generación de texto. Aunque tanto GPT como BERT son ejemplos de arquitecturas transformer que han estado influyendo en el campo del procesamiento de lenguaje natural en los últimos años, tienen diferentes fortalezas y debilidades que los hacen adecuados para diferentes tipos de tareas [23]. A diferencia de BERT, que se enfoca en entender el contexto de manera bidireccional para clasificación y predicción, GPT destaca en generar largas secuencias de texto con alta precisión, siendo especialmente útil en la creación literaria y diálogos.

En 2019 se introdujo GPT-2 [12], que es una evolución del modelo GPT original, potenciado con más parámetros para mejorar la generación textual. Se distingue por producir textos coherentes y lograr un rendimiento destacado en diversas tareas lingüísticas mediante el "few-shot learning", sin necesidad de entrenamiento específico por tarea. En términos generales, la estructura de un modelo como GPT-2 se describe de la siguiente manera 2.11:

- **Capa de Embedding:** Esta capa convierte las palabras ingresadas en vectores densos que el modelo puede interpretar y procesar.
- **Bloques de Atención Múltiple:** GPT-2 implementa varios bloques de atención para evaluar la relevancia de diferentes segmentos del texto ingresado cuando realiza predicciones.
- **Capa de Atención:** Esta capa contiene mecanismos de atención, donde las consultas (Q), claves (K) y valores (V) interactúan para determinar qué partes del texto son más relevantes en cada paso del procesamiento.
- **Feed Forward y Normalización (Add and Norm):** Después de cada bloque de atención, hay una capa de propagación hacia adelante y un proceso de normalización aplicado a cada token, y este conjunto de operaciones se repite en múltiples ocasiones.
- **Capa Lineal y Clasificador:** Al concluir el proceso, los vectores generados son procesados a través de capas lineales adicionales para tareas específicas, y finalmente, una capa clasificadora ofrece la salida definitiva del modelo.

GPT-3 [24], introducido en 2020, representa una expansión significativa sobre GPT-2, albergando 175 mil millones de parámetros, lo cual supone un incremento superior a 100 veces en tamaño en comparación con su antecesor. GPT-3 utiliza la misma arquitectura de modelo de transformador que GPT-2, pero ha sido escalado masivamente en tamaño y capacidad. Este avance en la magnitud del modelo ha propiciado mejoras destacables en su desempeño en diversas tareas de procesamiento del lenguaje natural. En particular, destaca en el aprendizaje a partir de pocos ejemplos, donde GPT-3 es capaz de ejecutar tareas con un número limitado de ejemplos ilustrativos o incluso sin ellos, tan solo mediante instrucciones en lenguaje coloquial.

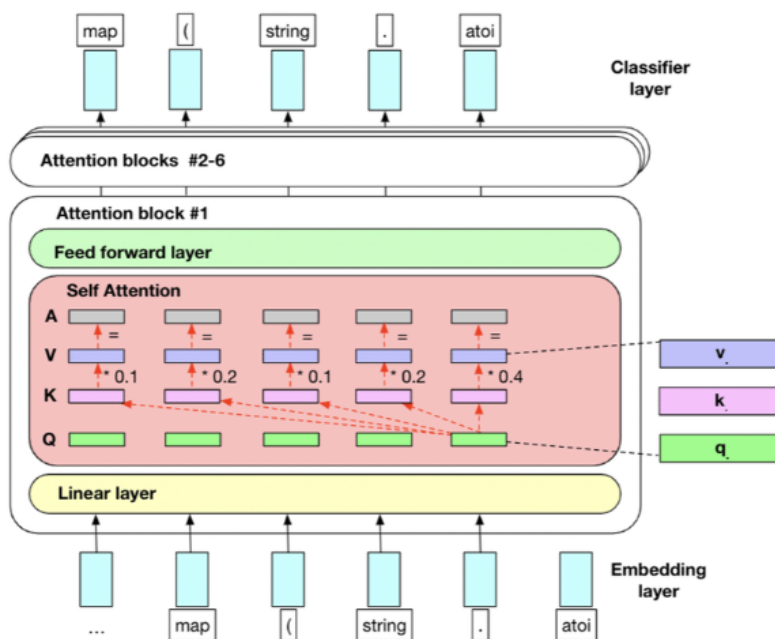


Figura 2.11: Ejemplo de arquitectura de GPT-2 [12]

Después de GPT-3, el ámbito de los modelos de lenguaje ha seguido avanzando a un ritmo acelerado. Un estudio de 2023 [25] examina la serie de modelos GPT-3, abarcando las novedades más recientes como ChatGPT y GPT-4, resaltando su creciente aceptación y utilidad en diversas actividades relacionadas con el procesamiento de lenguaje natural. Específicamente, GPT-4 se erige como una evolución en esta secuencia de desarrollo, proporcionando mejoras notables en aspectos como magnitud y habilidades de procesamiento lingüístico.

GPT-4 fue presentado por OpenAI en 2023 [25], y como se ha dicho anteriormente, es un avance notable sobre GPT-3, especialmente por su habilidad para manejar y analizar textos e imágenes, ampliando así sus posibles usos. Destaca por su rendimiento comparable al humano en diversos tests profesionales y académicos, además de mostrar un progreso significativo en precisión y en mantener conductas adecuadas después de su entrenamiento.

Desde el lanzamiento de GPT-4, el campo de los modelos generativos ha seguido evolucionando rápidamente. En 2024 y comienzos de 2025, OpenAI ha continuado perfeccionando sus modelos con nuevas versiones optimizadas de GPT-4, como GPT-4-turbo, que destaca por su mayor eficiencia, menor coste de inferencia y capacidad para mantener un contexto más extenso. Además, se ha mejorado la forma en que el modelo integra información de diversas fuentes, lo que le permite procesar imágenes, audio y documentos de una manera más natural y precisa. Estos avances han abierto la puerta a aplicaciones más sólidas en asistentes virtuales, diagnósticos automáticos, generación científica y educación personalizada, reafirmando el papel de los LLMs como herramientas versátiles en entornos profesionales y de investigación.

En 2024, ha surgido un nuevo modelo de lenguaje a gran escala especializado en el dominio biomédico llamado **BiomedLM** [26], desarrollado por investigadores del Allen Institute for AI. A diferencia de otros modelos mencionados anteriormente como BioBERT, que se basan en arquitecturas tipo encoder (como BERT), BioMedLM sigue una arquitectura autoregresiva basada únicamente en decodificadores, similar a la de los modelos GPT. Y se ha entrenado específicamente en grandes volúmenes de datos biomédicos, superando a modelos anteriores como PubMedBERT y BioGPT en tareas de comprensión, clasificación y respuesta en lenguaje natural biomédico.

BiomedLM se ha entrenado sobre más de 16 millones de artículos científicos de PubMed y otros datos biomédicos, y su arquitectura se basa en una versión optimizada de decoder-only transformers. La innovación más destacable de BiomedLM es su capacidad para razonar sobre texto médico con precisión contextual, lo que lo hace especialmente útil en tareas como generación de hipótesis, búsqueda de literatura médica relevante y soporte para decisiones clínicas.

En conclusión, el campo de los modelos de lenguaje de gran escala (LLM) está viviendo una evolución impresionante, con progresos notables y rápidos. BERT y GPT han sido pilares clave en nuestra habilidad para analizar y crear texto natural de forma efectiva. Estos modelos han abierto caminos para nuevas aplicaciones y optimizaciones en las ya existentes, estableciendo los cimientos para futuras innovaciones. Con cada año que transcurre, la comunidad de IA sigue superando las fronteras de lo conocido y lo técnico, refinando estos sistemas para que sean más ágiles, exactos y multifacéticos, allanando el camino para la próxima generación de avances en LLMs.

Desarrollo

En esta sección se describen de manera estructurada y rigurosa las distintas etapas seguidas para alcanzar el objetivo planteado en este trabajo. Se comienza con una exposición general de las fases que habitualmente conforman un proyecto de Deep Learning (DL), para, a continuación, detallar el desarrollo específico de cada una de ellas en el contexto del presente estudio, así como las decisiones técnicas adoptadas en cada caso. El proyecto se articula en varios componentes complementarios que, integrados de forma coherente, conforman la solución final propuesta.

3.1 Arquitectura del proyecto completo

En la Figura 3.1 se puede observar el resumen del flujo del proyecto que se detallará en las secciones siguientes paso a paso. Este esquema muestra de forma visual y resumida la estructura general del proyecto desarrollado.

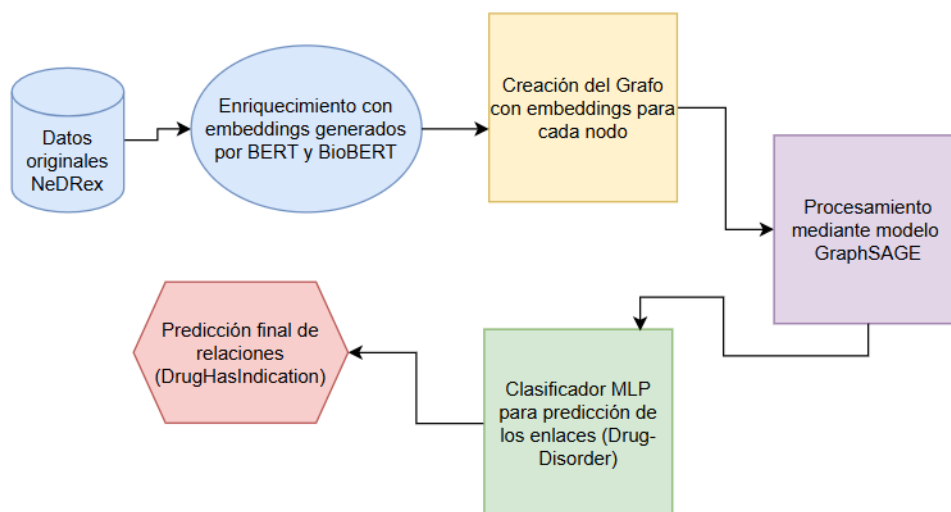


Figura 3.1: Flujo del proyecto completo

El proceso inicia con los datos originales extraídos de NeDRex, que constituyen la base del grafo inicial que se empleará para el aprendizaje. Estos datos se enriquecen posteriormente mediante embeddings generados con los modelos BERT y BioBERT para capturar información semántica adicional y transformar la información a una representación compacta para su uso. Tras esta etapa, se construye un grafo con estos embeddings enriquecidos, que posteriormente es procesado mediante el modelo de red neuronal de grafos GraphSAGE heterogéneo (u otra GNN que se seleccione) para obtener representaciones aún más profundas de los nodos.

Finalmente, estas representaciones se utilizan en un clasificador Multilayer Perceptron (MLP) específico para predecir la existencia de enlaces del tipo DrugHasIndication, que corresponden a la relación entre medicamentos y enfermedades. En la sección de resultados, se presentan y analizan detalladamente los resultados obtenidos con esta metodología en las diferentes partes del proyecto.

Es importante destacar que el objetivo principal de este proyecto es predecir enlaces entre medicamentos y enfermedades, específicamente aquellos del tipo DrugHasIndication. Una vez entrenado, el modelo toma como entrada un medicamento o una enfermedad y, a partir de esta entrada, devuelve un conjunto de enfermedades o medicamentos asociados, respectivamente. Junto con estas predicciones, el modelo proporciona un score que refleja la probabilidad de que cada enlace sea correcto.

Por ejemplo, si se introduce un medicamento, el modelo devolverá las enfermedades relacionadas con él, junto con el score que mide la fiabilidad de esa relación. De forma análoga, si se introduce una enfermedad, el modelo devolverá los medicamentos indicados para tratarla. Este enfoque permite recomendar tratamientos basados en enfermedades específicas o identificar medicamentos relevantes para determinados trastornos.

Este proyecto tiene un gran potencial, ya que puede ser especialmente útil en áreas como el reposicionamiento farmacológico y la identificación de nuevas relaciones entre medicamentos y enfermedades. Al generar una lista de entidades asociadas y sus scores, el sistema no solo proporciona información relevante, sino que también abre nuevas posibilidades en la investigación biomédica. Gracias al uso de un grafo enriquecido y modelos preentrenados como BioBERT, el modelo es capaz de descubrir relaciones complejas que, en muchos casos, podrían no ser tan evidentes. Esto representa una herramienta poderosa para encontrar nuevas conexiones en el ámbito de la salud, impulsando la innovación y contribuyendo a la mejora de los tratamientos médicos.

3.2 Fases en la creación de un proyecto de DL

La Figura 3.2 representa las partes más importantes de un proyecto de Deep Learning y se observa que se puede dividir en 5 fases principales. La primera fase sería la creación del dataset. En nuestro caso, el dataset va a ser los embeddings generados por un modelo LLM sobre la base de datos Nedrex. La segunda fase se centra en la preparación del dataset; seguidamente, se crea el modelo mediante su entrenamiento y validación, el testeo comprobará que el modelo es válido, y finalmente, este se desplegará para su uso. Se seguirá esta metodología en las siguientes secciones para el tratado de los datos y el procesamiento mediante las redes neuronales seleccionadas. Así mismo se describirá la forma de entrenamiento y testeo presentando en el capítulo siguiente todos los resultados obtenidos.

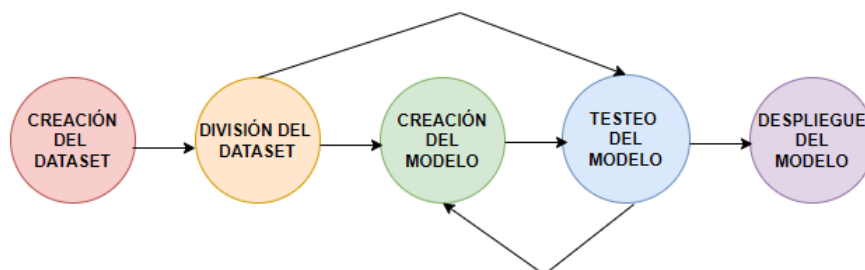


Figura 3.2: Fases en la creación de un proyecto de Deep Learning

3.3 Obtención del Dataset - NeDRex

NeDRex es una plataforma interactiva de medicina en red para la identificación de módulos de enfermedades y la reutilización de fármacos [27]. Esta será la base de datos (nuestro dataset) que se usará en este trabajo para poder identificar los fármacos correspondientes a determinadas enfermedades. Se descargará en formato csv y más adelante se detallará cómo se usa. La estructura de NeDRex se compone de una estructura de grafos. Está formada por diferentes nodos y sus caminos asociados (edges). Se puede observar en la imagen 3.3 cuáles son sus nodos y sus edges correspondientes.

En este proyecto nos vamos a centrar en los nodos de “Disorder” y “Drug” para poder predecir el camino “Drug has indication”. Ya que el objetivo del trabajo consiste en poder decirle a la red una determinada enfermedad y que proporcione los fármacos que podrían ser válidos para usarse con esa enfermedad determinada y viceversa. Los datos se han obtenido a través de la aplicación NeDRexApp [27], que está diseñada para usarse con Cytoscape [28], que es un software de código abierto diseñado para visualizar, analizar e interpretar redes complejas o grafos. Es ampliamente utilizado en investigación científica, especialmente en bioinformática y biología de sistemas, aunque también puede aplicarse a cualquier tipo de datos estructurados en red.

Una vez instalado Cytoscape, se instala NeDRexApp desde la propia AppStore de Cytoscape. Usando esta aplicación he creado mi propio grafo, como se puede ver en la siguiente imagen 3.4, simplificando algún nodo de la estructura original de NeDRex. A partir de este grafo, se han descargado dos archivos en formato CSV: uno con toda la información de los nodos y otro con las conexiones (edges). Estos archivos formarán el dataset que se usará para entrenar a la red en este proyecto. En la Tabla 3.1 se puede observar en detalle el tamaño de cada nodo y de cada edge que va a constituir el dataset del proyecto.

Para preparar los datos de cara a la siguiente fase del proyecto, se aplicó un proceso de limpieza utilizando un script en Python. Este paso es importante, ya que garantiza que los datos estén bien estructurados, clave para obtener embeddings precisos y efectivos, que puedan capturar correctamente las relaciones semánticas entre los nodos.

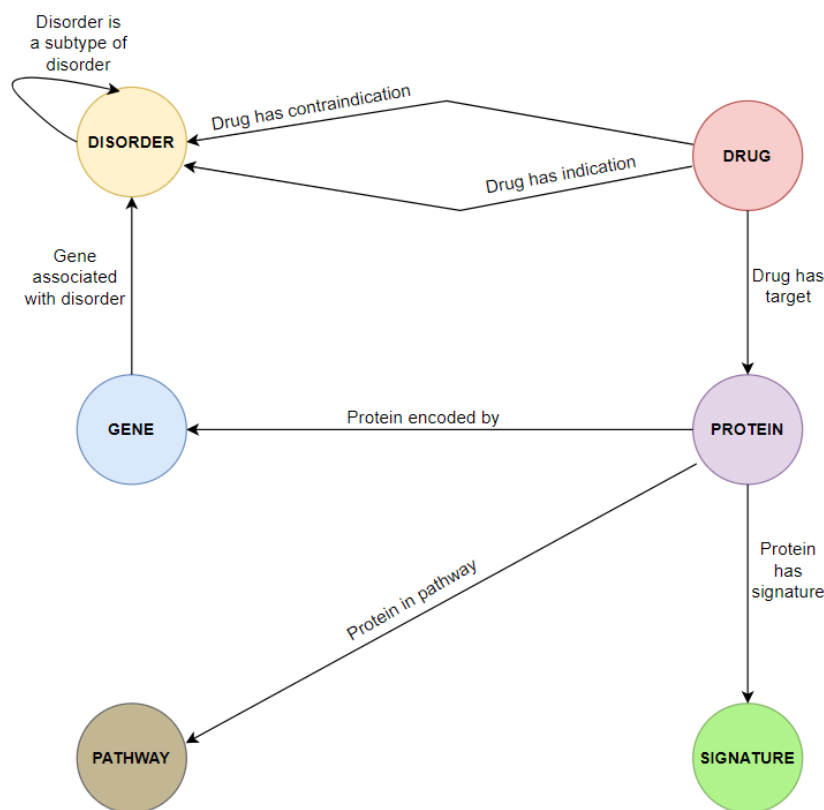


Figura 3.3: Estructura del grafo de NedRex completa

El proceso de limpieza se centró en dos aspectos principales:

- **Manejo de saltos de línea dentro de campos:** En muchos archivos CSV es común encontrar saltos de línea dentro de los campos, especialmente cuando estos campos están rodeados por comillas dobles. Estos saltos de línea pueden dividir erróneamente los campos, lo que dificulta el análisis posterior y afecta la integridad de los datos. Para resolver este problema, el script detecta las líneas divididas por saltos de línea dentro de los campos y las fusiona, asegurando que los campos que deberían permanecer intactos no se vean alterados.
- **Concatenación de líneas divididas:** El script también gestiona casos en los que los campos se extienden a varias líneas debido a los saltos de línea internos. Para ello, se utiliza un buffer que acumula las líneas hasta asegurar que un campo ha terminado. Este proceso se basa en el conteo de comillas dobles en cada línea, lo que indica si el campo sigue abierto. Una vez que el campo está cerrado (es decir, cuando el número de comillas dobles es par), se reemplazan los saltos de línea internos por espacios, evitando que se dividan incorrectamente los campos.

NODOS/EDGES	UNIDADES
Disorder	26015
Drug	7167
Gene	19394
Protein	30501
DrugHasIndication	17728
DrugHasTarget	30088
ProteinEncodedByGene	30432
DisorderIsSubtypeOfDisorder	77762
GeneAssociatedWithDisorder	48497

Tabla 3.1: Número de nodos y relaciones en el grafo biomédico.

Estas técnicas fueron elegidas porque el dataset original contenía múltiples casos de campos mal estructurados, con saltos de línea internos que interferían con la correcta lectura y manipulación de los datos. Si no se hubiese realizado este proceso de limpieza, la calidad de los embeddings generados podría haberse visto afectada, ya que los modelos de lenguaje requieren que los textos estén correctamente organizados para captar las relaciones semánticas entre los elementos del grafo. De esta forma, la limpieza de los datos permitió que los textos quedaran bien estructurados y listos para la siguiente fase del proyecto, en la que se generarán los embeddings.

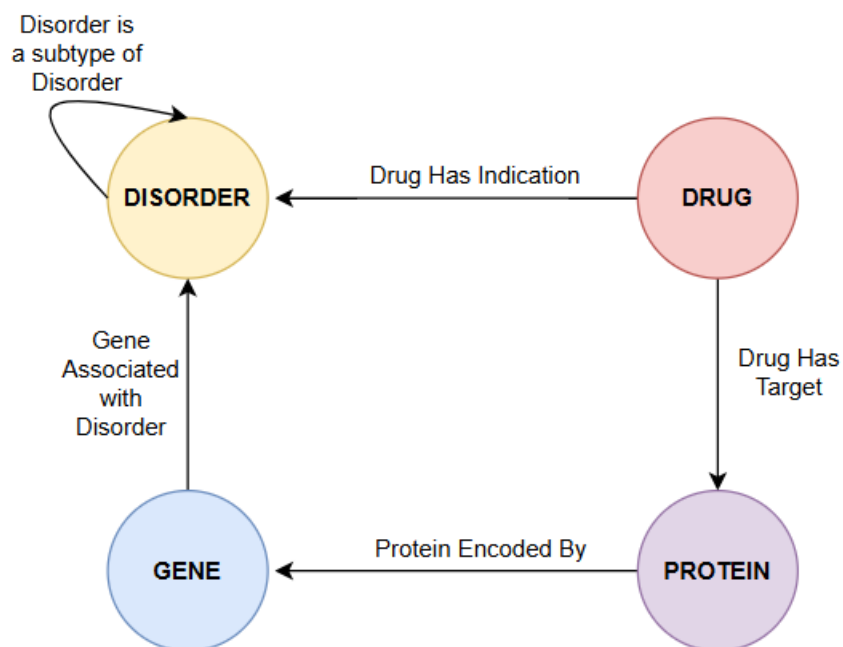


Figura 3.4: Estructura del grafo del proyecto

3.4 Preprocesado del Dataset - BERT y BioBERT

En este proyecto se ha decidido incorporar BERT, un modelo avanzado de procesamiento del lenguaje natural basado en transformers [9]. Esto se debe a que se busca mejorar la representación de los nodos del grafo mediante embeddings. Aunque el grafo inicial, construido con datos obtenidos desde NeDRexApp, proporciona información útil, ésta se limita principalmente a aspectos estructurales específicos del dataset original. Al utilizar embeddings generados con BERT, se pretende captar mejor la información semántica y contextual de cada nodo, permitiendo que la red de grafos aprenda patrones más completos y precisos. En resumen, usar BERT en este proyecto va a permitir capturar relaciones complejas y mejorar considerablemente la calidad y eficacia del análisis, ya que permite que la red neuronal aproveche información contextual adicional y, en consecuencia, mejore la precisión de sus predicciones, siendo esto muy importante en ámbitos biomédicos.

La elección del modelo LLM se basa en la estructura de los datos proporcionados, donde cada nodo contiene una serie de atributos que son interesantes para poder generar información adicional y dar un mayor contexto. Estos atributos proporcionan descripciones textuales que contienen información semántica clave sobre cada nodo, como pueden ser las características de una enfermedad, un medicamento o una proteína. Utilizar estos atributos como entrada para un modelo de lenguaje permite transformar el texto en embeddings numéricos que capturan relaciones complejas entre los nodos y enriquecen la capacidad de aprendizaje de la red neuronal de grafos.

Para ver mejor lo que nos puede aportar realizar estos embeddings, se presentan ejemplos de dos tipos de nodos: uno de tipo “Drug” y otro de tipo “Disorder”. Estos ejemplos muestran cómo las descripciones textuales de los nodos pueden ser procesadas para generar embeddings semánticos, aportando mucho más contexto y enriqueciendo los datos.

Ejemplo de nodo de tipo Drug:

- **displayName:** Aspirin
- **description:** A medication used to treat pain, fever, and inflammation.
- **drugCategories:** Nonsteroidal anti-inflammatory drug (NSAID)
- **synonyms:** Acetylsalicylic acid
- **type:** Drug

Ejemplo de nodo de tipo Disorder:

- **displayName:** Schizophrenia

- **description:** A major psychotic disorder characterized by abnormal thoughts, hallucinations, and delusions.
- **synonyms:** Schizophrenia, schizophrenia (disease), schizoaffective disorder
- **type:** Disorder

En estos ejemplos, los atributos como **description**, **drugCategories** y **synonyms** proporcionan información semántica clave sobre cada nodo. Esta información es crucial para entender las relaciones entre diferentes tipos de nodos, como las conexiones entre medicamentos y enfermedades.

La razón para utilizar estos atributos en la generación de embeddings es que los modelos de lenguaje, como BERT o BioBERT, pueden aprender representaciones vectoriales de estos textos. Estos embeddings permiten que la red neuronal de grafos capture las relaciones semánticas subyacentes entre los nodos, mejorando la capacidad del modelo para identificar enlaces relevantes, como el enlace “DrugHasIndication”.

Dentro de los LLM, el elegido para este proyecto, como se ha mencionado anteriormente, ha sido BERT, conocido por su capacidad para comprender el contexto de las palabras dentro de una oración de manera bidireccional. Como se explicó en la sección del Estado del Arte, BERT es uno de los modelos más influyentes en el ámbito del procesamiento del lenguaje natural (NLP) debido a su diseño basado en la arquitectura Transformer, que le permite capturar dependencias a largo plazo y generar representaciones robustas de texto.

En el caso de este trabajo, BERT se ha utilizado en su variante base (“bert-base-uncased”), que ha sido pre-entrenado utilizando un conjunto amplio y diverso de textos en lenguaje natural. Este modelo se aplicó directamente a las descripciones textuales de los nodos, generando embeddings de 768 dimensiones. Estos embeddings codifican tanto el significado semántico como el contexto de las descripciones, lo que resulta especialmente útil para identificar similitudes entre nodos o inferir relaciones entre ellos.

El flujo de trabajo para generar los embeddings consiste en procesar los atributos textuales mediante el tokenizador de BERT, que convierte el texto en tokens y agrega indicadores especiales como [CLS] y [SEP] para delimitar las secuencias. Posteriormente, estas secuencias tokenizadas se introducen en el modelo BERT, y los embeddings se extraen de la capa final correspondiente al token [CLS]. Este token se utiliza porque su embedding está diseñado para capturar una representación global de la secuencia de entrada.

Una vez generados, los embeddings se asignan a cada nodo en el grafo, sustituyendo o complementando los atributos numéricos originales. De esta manera, la red neuronal de grafos no solo aprende de las conexiones estructurales del grafo, sino también de las características semánticas codificadas en los embeddings de las descripciones.

Tras evaluar los resultados obtenidos con BERT, que se detallarán en el apartado de resultados, se observó que existía margen de mejora, especialmente considerando la naturaleza biomédica de los datos. Por ello, se decidió incorporar también BioBERT, un modelo especializado para textos biomédicos [29]. Este modelo está preentrenado con literatura biomédica, lo que permite capturar mejor la terminología específica y el contexto del dominio médico y biológico.

De esta manera, el uso de Bert o Biobert, permite mejorar la precisión en la representación semántica y contextual de cada nodo, permitiendo que la red neuronal de grafos aprenda patrones más completos y específicos.

En conclusión, la inclusión de embeddings preentrenados a nuestro proyecto tiene varias ventajas. En primer lugar, permite que el modelo aproveche conocimiento previo almacenado en BERT, lo que es particularmente valioso en escenarios donde las relaciones explícitas entre nodos son limitadas. En segundo lugar, mejora la eficiencia computacional, ya que los embeddings precomputados pueden ser reutilizados en lugar de requerir que la red aprenda todas las representaciones desde cero, proporcionando un marco versátil para explorar datos relacionados con enfermedades, medicamentos y otros conceptos biomédicos. Finalmente, se mejora la generalización del modelo, ya que los embeddings capturan patrones lingüísticos que pueden no estar presentes directamente en los datos estructurales del grafo.

Este enfoque híbrido combina el poder de los modelos de lenguaje preentrenados con la flexibilidad de las redes neuronales de grafos, ofreciendo un marco robusto para analizar datos heterogéneos como los presentes en NeDRex.

A lo largo del proyecto, se observará cómo esta integración afecta positivamente al rendimiento de la red, destacando el impacto de los embeddings basados en lenguaje natural en tareas de aprendizaje supervisado en grafos, maximizando así el potencial del modelo para resolver problemas complejos en el ámbito biomédico.

3.5 Creación del modelo Graph Neural Network (GNN)

En esta sección se aborda la esencia de nuestro proyecto, las Redes Neuronales de Grafos (GNNs), fundamentales para el desarrollo de este trabajo.

Las GNNs son estructuras de redes robustas diseñadas para almacenar y procesar una gran cantidad de información a través de nodos y aristas en un grafo. La generación de representaciones vectoriales o embeddings de baja dimensión dentro de estos grafos permite transformar información compleja en un formato más simple, favoreciendo la eficiencia computacional y la posibilidad de encontrar soluciones comparables a problemas similares ya enfrentados en el ámbito del aprendizaje automático profundo. La habilidad de las GNNs para sintetizar y actualizar estos embeddings es fundamental para su aplicación práctica.

Una característica distintiva de las GNNs es el proceso de **message passing** [13], donde se intercambian mensajes en forma de vectores entre nodos, lo cual permite la actualización dinámica de los embeddings basados en la información más reciente de los nodos vecinos. Este proceso refleja un enfoque iterativo en el que cada nodo del grafo refina su estado a través de múltiples ciclos de actualización, permitiendo que el modelo capture con mayor precisión las estructuras y patrones subyacentes en los datos.

El principal desafío al modelar datos estructurados en grafos es capturar efectivamente las dependencias e interacciones entre los nodos. Las técnicas tradicionales de aprendizaje profundo, como las redes neuronales convolucionales para imágenes o las redes neuronales recurrentes para secuencias, no son directamente aplicables a los grafos. Los grafos carecen de la estructura de malla inherente de las imágenes o del orden lineal de las secuencias.

El Paso de Mensajes Neutrales (Message Passing) es una técnica clave en las Redes Neuronales de Grafos (GNN) que permite modelar cómo interactúan los nodos entre sí dentro de un grafo. Cada nodo intercambia información con los nodos vecinos, lo que le ayuda a actualizar su propia representación basándose tanto en la estructura del grafo como en las características de los nodos. Este proceso es fundamental para que las GNNs puedan capturar información de los nodos no solo desde una perspectiva local, sino también global, lo que les permite generar representaciones detalladas y completas de los nodos. Esta habilidad para modelar dependencias complejas hace que el Paso de Mensajes Neutrales sea especialmente útil en tareas como la predicción de enlaces, la clasificación de nodos y grafos, y la recomendación, posicionándose como una herramienta poderosa en el aprendizaje automático aplicado a grafos.

En cada iteración del paso de mensajes en una GNN, se actualiza una incrustación oculta $h_u^{(k)}$ para cada nodo u , basada en la información recopilada de su grafo vecino $N(u)$. Esta actualización puede expresarse de la siguiente manera:

$$h_u^{(k+1)} = \text{ACTUALIZAR}^{(k)} \left(h_u^{(k)}, \text{AGREGAR}^{(k)} \left(\{h_v^{(k)} \mid v \in N(u)\} \right) \right) \quad (3.1)$$

Aquí, **Actualizar** y **Agregar** son funciones diferenciables arbitrarias, típicamente implementadas como capas de redes neuronales de neuronas. La función **Agregar** toma los embeddings de los nodos en el vecindario $N(u)$ y genera un mensaje $m_{N(u)}^{(k)}$ basado en esta información agregada. La función **Actualizar** combina el mensaje $m_{N(u)}^{(k)}$ con el embedding previo $h_u^{(k-1)}$ para producir el embedding actualizado $h_u^{(k)}$. Los embeddings iniciales en la iteración $k = 0$ se establecen en las características de entrada x_u para todos los nodos u . Después de K iteraciones de paso de mensajes, podemos usar la salida de la última capa para definir los embeddings para cada nodo z_u .

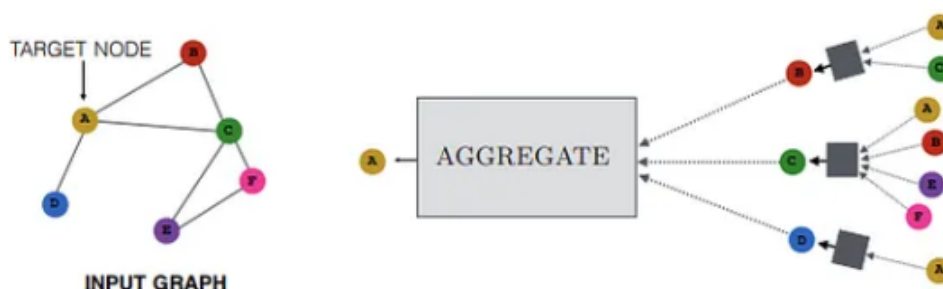


Figura 3.5: Proceso de agregación en una Red Neuronal de Grafos [13]

3.5.1 GraphSAGE

El modelo utilizado en este trabajo se basa en una arquitectura GraphSAGE (Graph Sample and Aggregate) [4] adaptada para datos heterogéneos. Como se ha descrito en la sección del Estado del Arte, esta GNN fue propuesta por William L. Hamilton, Rex Ying y Jure Leskovec. Es un método inductivo para el aprendizaje en grafos que permite generalizar a grafos no vistos durante el entrenamiento. Su objetivo principal es procesar los nodos y aristas de un grafo para generar representaciones (embeddings) que permitan predecir relaciones entre diferentes tipos de nodos, como las conexiones entre medicamentos y enfermedades.

Este enfoque resulta particularmente útil en grafos biológicos complejos, donde los nodos representan entidades heterogéneas (medicamentos, enfermedades, proteínas, etc.) y las relaciones entre ellos capturan interacciones biológicas relevantes.

Antes de entrar en el detalle sobre cómo opera GraphSAGE (SAmple y aggreGatE) y su arquitectura, es importante mencionar que existen dos métodos principales al abordar los grafos. Primero, el enfoque inductivo, que permite que el grafo varíe entre las fases de entrenamiento y predicción sin requerir que sea el mismo. En segundo lugar, el enfoque transductivo, donde el grafo debe mantenerse constante tanto durante el entrenamiento como en la predicción [30]. Podemos observar estos dos enfoques en la siguiente figura 3.6, en la que se puede ver que el enfoque transductivo se representa en la figura de arriba y el inductivo en la figura de la derecha, en la que el grafo puede variar.

Por estas razones se ha decidido utilizar esta red para el desarrollo del trabajo, ya que representa una innovación clave en el campo del aprendizaje profundo para grafos, destacándose por su habilidad para generar representaciones de nodos de forma inductiva.

La mayoría de las metodologías propuestas para crear embeddings de nodos en grafos antes de GraphSAGE exigían que todos los nodos estuvieran presentes durante la inicialización del modelo. Sin embargo, GraphSAGE permite que los modelos generalicen a nuevos nodos sin

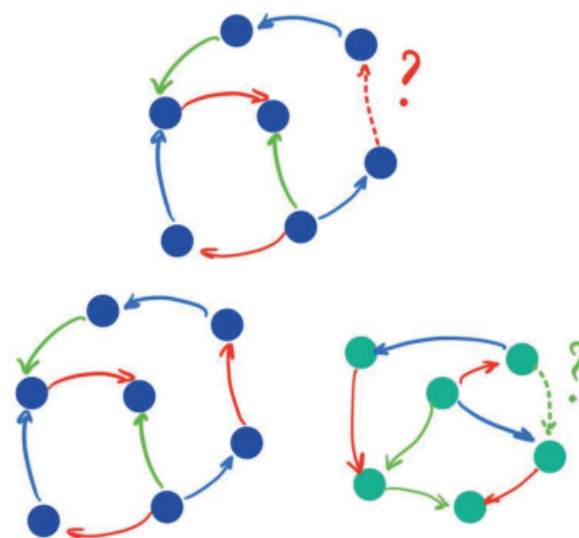


Figura 3.6: Enfoque transductivo vs inductivo [4]

necesidad de reentrenar completamente el sistema, proponiendo un método eficiente y escalable para la generación de embeddings mediante el uso de técnicas inductivas. Esto se logra a través de un mecanismo que genera embeddings utilizando únicamente la información del entorno inmediato del nodo, en lugar de requerir un embedding específico por nodo previamente entrenado.

Los métodos dominantes durante la popularización de las GNNs eran predominantemente transductivos. Estos métodos se centraban en generar una serie de embeddings, no en generar funciones capaces de hacerlo de manera autónoma. Se les conocía como métodos de shallow embedding debido a su incapacidad para generalizar más allá de los nodos vistos durante el entrenamiento.

Esta limitación, inherente a los enfoques puramente transductivos, los hacía computacionalmente ineficientes y no permitía aprovechar plenamente las características inherentes de los nodos. Este problema se aborda y se supera con técnicas como las empleadas en GraphSAGE.

GraphSAGE introduce una estrategia para generar embeddings representativos mediante el muestreo de vecinos para generar los embeddings, agregando y actualizando la información en sucesivas iteraciones. Esto se realiza a través de un proceso de paso de mensajes, y para lograrlo, se entrenan series de funciones de agregación que se optimizan con el tiempo. Al final del proceso, se obtienen embeddings que incorporan tanto el contexto local como información global, tal como se ilustra en la figura 3.7. En el que se puede observar que en el primer paso se muestrea el vecindario del nodo y en el segundo y tercer paso se utilizan las funciones de agregación para asimilar la información muestreada y predecir.

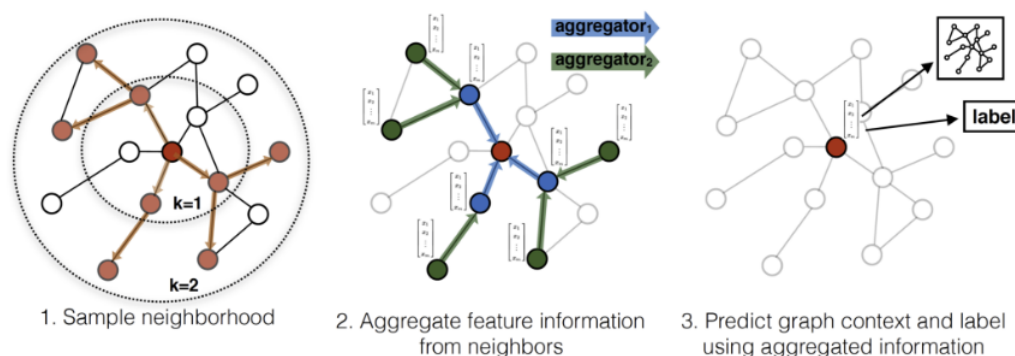


Figura 3.7: Pasos en el proceso de GraphSAGE [4]

3.5.2 Implementación del Modelo GraphSAGE

A continuación se detalla cómo se ha implementado este modelo Graph SAGE para resolver el problema de predicción de enlaces en grafos heterogéneos. El modelo está compuesto principalmente por tres partes:

- **GNN (Graph Neural Network)**

Esta parte del modelo utiliza capas convolucionales tipo SAGEConv para construir embeddings o representaciones enriquecidas de los nodos. Estas capas, acompañadas de técnicas como BatchNorm, funciones de activación no lineales (ReLU) y dropout, permiten que cada nodo capture información relevante de su entorno cercano en el grafo, mejorando así la capacidad del modelo para aprender patrones estructurales.

- **Clasificador (ClassifierMLP):**

Una vez que se obtienen las representaciones de los nodos, se utiliza un perceptrón multicapa (MLP) para predecir si existe o no un enlace entre ellos. En este paso, los embeddings de los nodos involucrados (en este caso, medicamentos y enfermedades) se concatenan primero, y luego la concatenación de estos embeddings se pasa a través del MLP para evaluar la compatibilidad entre los nodos. Es importante destacar que la capa MLP no realiza la concatenación, sino que simplemente procesa los embeddings concatenados, lo que permite al modelo aprender de las relaciones entre los pares de nodos.

- **Modelo completo (Model):**

Este módulo integra los dos anteriores y los adapta al contexto heterogéneo del grafo. Para ello, cada tipo de nodo tiene una capa lineal específica que transforma sus características originales (por ejemplo, embeddings generados previamente por modelos como

BERT o BioBERT) en representaciones comunes. Además, mediante la adaptación automática proporcionada por la función `to_hetero`, el modelo puede gestionar diferentes tipos de nodos y relaciones simultáneamente, asegurando así que sea compatible con estructuras complejas y variadas.

En la figura 3.8 se puede observar la arquitectura detallada de la implementación de la red neuronal de grafos, GraphSAGE, en el proyecto.

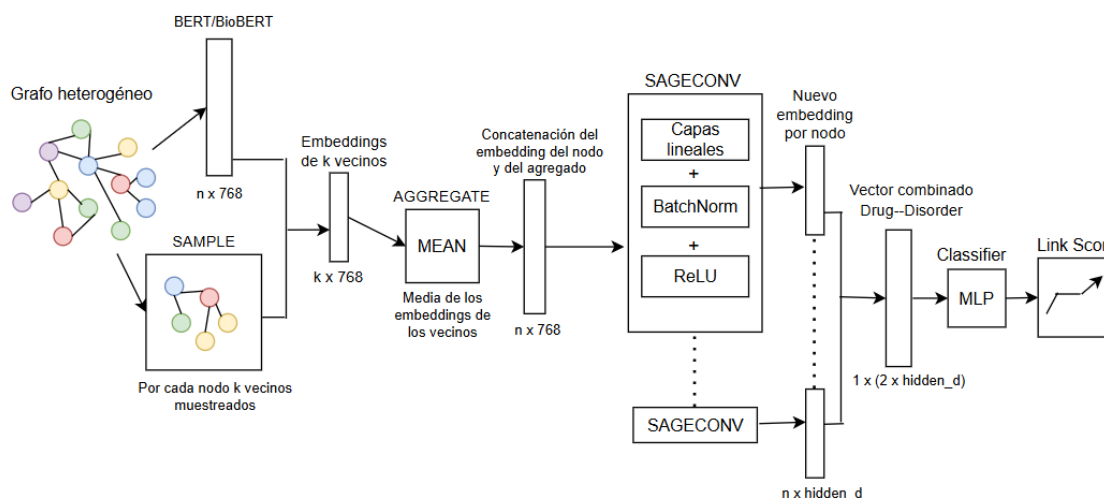


Figura 3.8: Arquitectura detallada de la implementación de GraphSAGE

En resumen, se ha implementado una arquitectura modular de GraphSAGE, que incluye inicialización de capas de convolución, procesamiento de características de nodo y predicción basada en productos escalares de embeddings. Al ser altamente modular y extensible, facilita la incorporación de nuevos tipos de nodos o relaciones. Además, la combinación de embeddings preentrenados de BERT con la agregación jerárquica de GraphSAGE permite capturar tanto las características textuales como las estructurales del grafo. Y la arquitectura heterogénea garantiza que el modelo pueda procesar datos complejos y diversos, una característica clave en aplicaciones biomédicas.

También es importante mencionar que el número de capas y parámetros utilizados en este modelo podrá ajustarse posteriormente según los resultados obtenidos, buscando así optimizar el rendimiento y mejorar la precisión en la tarea específica del proyecto.

Este enfoque proporciona una base sólida para abordar problemas de predicción en grafos biológicos, permitiendo identificar relaciones relevantes entre entidades como enfermedades, medicamentos y proteínas. En futuras secciones del documento, veremos los resultados que se han obtenido al aplicar este modelo.

3.5.3 Graph Attention Networks (GAT)

Además de utilizar la red GraphSAGE, se probará una configuración con otra red neuronal de grafos, basada en Graph Attention Networks (GAT) [6] para testear el impacto en los resultados y poder realizar una comparación posterior.

A diferencia de GraphSAGE, que utiliza un enfoque basado en el muestreo y la agregación de vecinos, GAT introduce el concepto de atención en el aprendizaje de representaciones de nodos. El mecanismo de atención en GAT permite que cada vecino de un nodo en el grafo tenga un peso dinámico, determinado por una “atención” que indica la importancia relativa de ese vecino para el nodo objetivo. Este proceso es realizado de manera automática, lo que otorga al modelo mayor flexibilidad para aprender representaciones basadas en las relaciones de los nodos más relevantes, sin necesidad de tener una estructura fija de agregación como en GraphSAGE.

El modelo de GAT emplea capas de convolución de atención, donde los nodos pueden asignar diferentes pesos a sus vecinos mediante una función de atención que se puede entrenar. De este modo, cada nodo puede “atender” de manera diferente a sus vecinos, lo que puede tener gran utilidad en grafos heterogéneos, como el que estamos utilizando, en el que los nodos pueden tener diferentes tipos y las relaciones pueden variar en importancia.

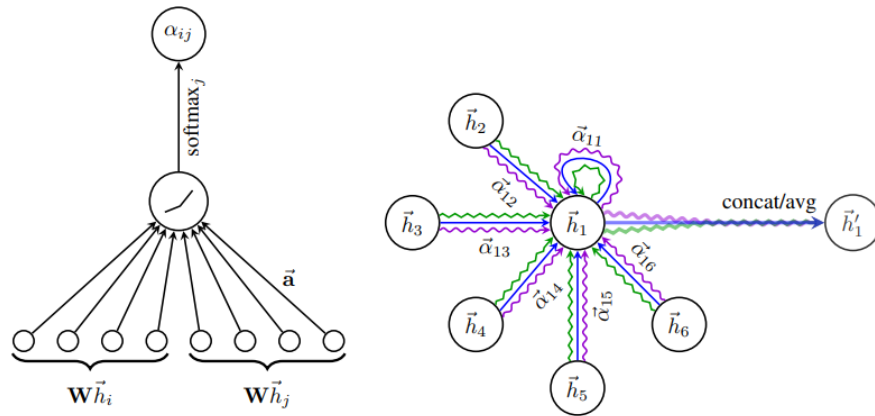


Figura 3.9: Esquema del mecanismo de atención en GAT. Cada nodo pondera las representaciones de sus vecinos mediante coeficientes de atención α_{ij} y genera una nueva representación combinando la información más relevante[6].

Para ilustrar este mecanismo, en la figura 3.9 se muestra un esquema visual simplificado del funcionamiento de la atención en GAT. Cada nodo calcula un coeficiente de atención α_{ij} con respecto a cada uno de sus vecinos, lo que permite ponderar de forma diferente la información que recibe de cada uno. Posteriormente, las representaciones de los vecinos se combinan mediante una operación de concatenación o media, generando una nueva representación del nodo que incorpora dicha atención.

3.5.4 Implementación del Modelo GAT

A continuación se detalla cómo se ha implementado este modelo para resolver el problema de predicción de enlaces en grafos heterogéneos. El modelo está compuesto principalmente por tres partes:

- **GNN (Graph Neural Network)**

El modelo GAT utiliza la capa de convolución GATConv para calcular los embeddings de los nodos. A diferencia de otros modelos de redes neuronales de grafos, GAT introduce el mecanismo de atención, permitiendo que cada vecino de un nodo reciba un peso diferente en función de su relevancia. Este peso es aprendido durante el entrenamiento, lo que permite que el modelo determine cuáles son los nodos más importantes en su vecindad. En nuestra implementación, se utilizan dos capas de atención: una primera capa con múltiples cabezas de atención, seguida de una segunda capa con una única cabeza para reducir la dimensionalidad de los embeddings.

- **Clasificador (EdgeDecoder)**

Una vez que se han calculado los embeddings de los nodos mediante el GAT, la siguiente etapa es la predicción de los enlaces entre nodos. Para esto, se utiliza un EdgeDecoder, que toma los embeddings de los nodos fuente y destino de un enlace y los concatena. Luego, pasa esta concatenación a través de una capa lineal que produce una puntuación, la cual se interpreta como la probabilidad de que exista un enlace entre esos dos nodos. Este mecanismo es crucial para la tarea de predicción de enlaces, ya que permite evaluar la interacción entre nodos en función de sus representaciones aprendidas.

- **Modelo completo (Model)**

El modelo completo integra tanto el GATEncoder (la parte de convolución) como el EdgeDecoder (la parte de predicción de enlaces). Durante el entrenamiento, el modelo recibe los embeddings de los nodos, al igual que antes se usa BERT y bioBERT para generarlos, y las conexiones entre ellos a través de `edge_index` (el índice de los enlaces en el grafo). Luego, realiza la predicción de enlaces entre nodos basándose en sus embeddings calculados. El modelo es capaz de manejar grafos heterogéneos de forma eficiente gracias a la función `to_hetero`, que permite gestionar múltiples tipos de nodos y relaciones de forma simultánea. Esta adaptación es fundamental para que el modelo pueda trabajar con grafos complejos, como los biológicos, que incluyen diferentes tipos de entidades y sus interacciones.

En la figura 3.10 se puede observar la arquitectura detallada de la implementación de la red neuronal de grafos, GAT, en el proyecto. En resumen, el modelo Graph Attention Network (GAT) es útil en tareas de predicción de enlaces en grafos heterogéneos debido a su capacidad para asignar diferentes pesos a los vecinos de un nodo, lo que le permite aprender relaciones

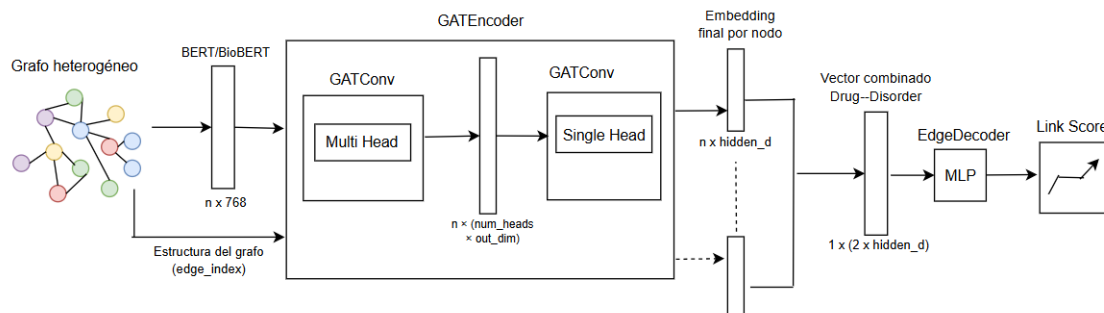


Figura 3.10: Arquitectura detallada de la implementación de GAT

más relevantes entre nodos. Este enfoque es especialmente ventajoso cuando algunos vecinos tienen más relevancia que otros. Sin embargo, este mecanismo de atención introduce una mayor complejidad computacional, ya que requiere calcular y optimizar los coeficientes de atención para cada par de nodos, lo que puede aumentar el tiempo de entrenamiento y el uso de memoria en comparación con modelos como GraphSAGE.

Por otro lado, GraphSAGE destaca por su eficiencia y escalabilidad, especialmente en escenarios donde los grafos pueden cambiar o expandirse con el tiempo. Su capacidad para generalizar a nodos no vistos durante el entrenamiento es una de sus principales ventajas. Aunque GAT es capaz de modelar relaciones más complejas entre nodos, su coste computacional lo hace menos adecuado para escenarios de generalización inductiva en grafos muy grandes.

Ambos modelos se utilizarán en este trabajo para comparar métricas y analizar los resultados con mayor detalle.

3.6 Entrenamiento del modelo GNN

Una vez definido el modelo basado en GraphSAGE o GAT, se tiene que entrenar para poder conseguir los objetivos que se persiguen en este proyecto. Para ello, se seleccionan y ajustan cuidadosamente diferentes hiperparámetros, como el número de epochs (iteraciones completas sobre todos los datos de entrenamiento), la tasa de aprendizaje (learning rate)...así como el optimizador elegido y sus parámetros específicos.

Para realizar el entrenamiento de la red se ha dividido el dataset en tres conjuntos: entrenamiento (70 %), validación (10 %) y test (20 %). Dado que este proyecto aborda un problema de predicción de enlaces (link prediction), la división en los conjuntos mencionados se realiza únicamente sobre las aristas del tipo "DrugHasIndication". Esto significa que, mientras los enlaces específicos se reparten entre los conjuntos de entrenamiento, validación y test, los no-

dos y los embeddings correspondientes permanecen sin cambios y son compartidos por todos los conjuntos. Esto se debe a que en problemas de link prediction interesa evaluar si el modelo es capaz de identificar enlaces nuevos o no observados entre nodos ya conocidos. De esta forma, el modelo aprovecha toda la información estructural y de contexto disponible durante el entrenamiento, facilitando que aprenda patrones eficaces para predecir correctamente las conexiones faltantes.

Se han implementado dos enfoques diferentes para entrenar y evaluar la red neuronal de grafos y diferentes hiperparámetros para poder ir mejorando la precisión de los resultados:

- **Enfoque basado en la pérdida BCE (Binary Cross-Entropy):**

En este caso, se utiliza la pérdida estándar BCE con una ponderación específica para compensar el desbalance existente entre clases positivas y negativas. Esta función de pérdida mide la diferencia entre las etiquetas reales y las probabilidades predichas por el modelo. Se define mediante la siguiente fórmula [31]:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)] \quad (3.2)$$

donde y_i representa la etiqueta real del enlace (1 para positivo, 0 para negativo), y $\hat{y}_i$ es la probabilidad predicha para dicho enlace.

Además de las métricas clásicas como la AUC (área bajo la curva ROC) y la precisión (accuracy), también se ha implementado la métrica recall@k. Esta métrica permite medir específicamente cuántos enlaces reales (positivos) se encuentran dentro de los k enlaces con mayor probabilidad predicha por el modelo, proporcionando una idea clara sobre la capacidad del modelo para identificar correctamente enlaces relevantes.

- **Enfoque basado en Margin Ranking Loss:**

Tras evaluar los resultados del enfoque BCE, se decidió también explorar la pérdida Margin Ranking Loss, que se centra en asegurar que los enlaces positivos sean rankeados consistentemente por encima de los negativos. En lugar de centrarse únicamente en una interpretación probabilística, este enfoque enfatiza la importancia del orden relativo entre enlaces. La función penaliza los casos en los que un enlace negativo tiene un valor mayor que uno positivo, y se define con la siguiente fórmula [32]:

$$\mathcal{L}_{\text{MRL}} = \frac{1}{N} \sum_{i=1}^N \max(0, \gamma - \hat{y}_i^+ + \hat{y}_i^-) \quad (3.3)$$

donde $\hat{y}_i^+$ y $\hat{y}_i^-$ son las puntuaciones predichas para los enlaces positivos y negativos respectivamente, y γ es un margen que define la separación deseada entre ambos. Este tipo de pérdida es especialmente útil en contextos en los que interesa priorizar los enlaces más relevantes desde un punto de vista clínico o práctico. Además, este enfoque

incorpora métricas avanzadas como el Recíproco del Rango Medio local (MRR) y la Ganancia Acumulada Descontada Normalizada (NDCG), que evalúan específicamente la calidad del ranking realizado por el modelo por cada fármaco, proporcionando así información más granular y valiosa sobre el desempeño del modelo. Estas métricas se definen a continuación con las siguientes fórmulas [33]:

$$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \quad (3.4)$$

$$\text{NDCG@k} = \frac{1}{\text{IDCG@k}} \sum_{i=1}^k \frac{2^{\text{rel}_i} - 1}{\log_2(i + 1)} \quad (3.5)$$

donde rank_i es la posición del primer elemento relevante para una consulta (por ejemplo, un medicamento), y rel_i es la relevancia del elemento en la posición i dentro del ranking.

Ambos enfoques cuentan con mecanismos adicionales como reducción dinámica de la tasa de aprendizaje (ReduceLROnPlateau), early stopping para evitar caer en overfitting y guardado automático del mejor modelo obtenido durante el entrenamiento según las métricas de validación, asegurando así la robustez y eficiencia del proceso de entrenamiento.

3.7 Testeo y despliegue del modelo

Esta es la última fase importante en el desarrollo y evaluación del modelo basado en Graph SAGE o GAT para la predicción del enlace "DrugHasIndication" (link prediction). El objetivo principal es comprobar que el modelo puede generalizar correctamente sobre enlaces nuevos que no haya visto antes, usando para ello el conjunto de test, que incluye enlaces completamente desconocidos para el modelo, aunque los nodos y embeddings sean los mismos que ya ha visto previamente.

Durante esta fase se evalúa si el modelo es capaz de identificar correctamente los enlaces reales y si cumple los criterios de precisión y calidad del ranking que se habían establecido como objetivos del proyecto. Para medir el rendimiento se utilizan métricas específicas como la AUC, el recall@k y otras métricas de ranking locales (MRR y NDCG), que nos permiten comprobar si el modelo es capaz de priorizar correctamente los enlaces más relevantes. Los resultados de estas métricas se detallarán más adelante en la sección de Resultados.

Además, para monitorizar el proceso de entrenamiento y comprobar su estabilidad, se representan las curvas de pérdida (loss) a lo largo de los epochs. Estas curvas permiten visualizar si la pérdida disminuye de manera consistente durante el entrenamiento y si el modelo está aprendiendo de forma progresiva sin caer en overfitting.

Asimismo, en la sección de resultados se incluirán en forma de tablas los resultados de la predicción, en los que se muestra para cada enfermedad el conjunto de los cinco medicamentos más importantes que el modelo predice, junto con el score correspondiente. Del mismo modo, se realizará el análisis inverso, en el que se muestra para cada medicamento, las enfermedades que el modelo considera relevantes. Estos resultados permitirán interpretar de forma más clara la utilidad práctica del modelo, así como identificar patrones clínicamente relevantes y posibles oportunidades de reposicionamiento farmacológico.

En caso de que los resultados obtenidos no sean satisfactorios, será necesario realizar ajustes adicionales en el entrenamiento del modelo. Estos ajustes pueden consistir en cambiar hiperparámetros, aumentar o mejorar la calidad del conjunto de datos, modificar la estructura del modelo o incluso plantearse usar otro enfoque distinto, garantizando así que el modelo final sea robusto, fiable y adecuado para aplicaciones prácticas o futuras investigaciones sobre este tema.

3.8 Implementación en GitHub

Para dar visibilidad y de cara a poder comprobar más en detalle las implementaciones específicas detalladas en esta sección se provee del enlace al GitHub con todo el proyecto. En él se pueden encontrar diferentes notebooks en Python con los procesos de limpieza y adecuación del dataset, y los procesos de entrenamiento y evaluación de las redes neuronales empleadas.

<https://github.com/GemaMarin/advanced-medical-recommendation-gnn-llm>

Resultados

En esta sección se presentan y analizan detalladamente los resultados obtenidos en las diferentes fases y enfoques implementados en este trabajo. Se explicarán las métricas que se han utilizado para medir estos resultados de forma cuantitativa, y así poder valorar objetivamente el rendimiento del modelo con las diferentes configuraciones.

- Comparación de los resultados obtenidos utilizando diferentes métodos para generar embeddings (BERT vs BioBERT).
- Comparación de dos modelos de GNN (GraphSAGE y GAT).
- Comparación de dos métodos de entrenamiento diferentes (Binary Cross-Entropy vs Margin Ranking Loss).

Además, en el último apartado se mostrarán predicciones concretas realizadas por el modelo para el enlace “DrugHasIndication”, ya sea dando una enfermedad y que recomiende medicamentos o viceversa, seleccionando un medicamento y que nos muestre las enfermedades en las que este medicamento podría ser útil.

4.1 Métricas de evaluación para LLM

Para evaluar de manera rigurosa la calidad de los embeddings generados mediante modelos de lenguaje como BERT [11] y BioBERT [21], es esencial disponer de métricas específicas que permitan cuantificar de forma objetiva cómo de bien capturan estos embeddings las relaciones semánticas que están presentes en los datos biomédicos que se han utilizado. Por tanto, se va a realizar una evaluación intrínseca, es decir, analizar directamente los embeddings. Esto no depende de una tarea específica, sino que analiza qué tan bien capturan relaciones semánticas.

Una métrica especialmente relevante para este tipo de evaluación es **el Coeficiente de Correlación de Spearman** [34]. Esta métrica permite comprobar si existe una relación consistente entre dos variables, es decir, si al aumentar una también tiende a aumentar (o disminuir) la otra, aunque no sea de forma exacta. En nuestro contexto, Spearman nos ayuda a evaluar si la similitud entre los nodos, que se mide mediante la similitud del coseno entre sus embeddings, se corresponde con relaciones reales y validadas en el ámbito biomédico, como las de tipo DrugHasIndication. El coeficiente de Spearman toma valores en el rango de -1 a +1, donde:

- Un valor próximo a +1 indica una fuerte correlación positiva, es decir, pares de nodos relacionados clínicamente tienden a estar representados por embeddings más cercanos

semánticamente. Esta cercanía semántica se mide mediante la similitud del coseno, que calcula el ángulo entre los vectores de embeddings de los nodos. Cuanto menor es el ángulo entre ellos (es decir, cuanto más alta es la similitud del coseno), más estrechamente relacionados están los nodos en términos semánticos.

- Un valor cercano a -1 implica lo contrario, indicando que los embeddings no logran captar adecuadamente las relaciones de interés, asignando menor similitud a nodos relacionados que a pares aleatorios.
- Un valor cercano a 0 indica una falta de correlación, lo que significaría que los embeddings no reflejan la estructura semántica real de los datos biomédicos.

Además del coeficiente de correlación de Spearman, otras técnicas complementarias pueden ayudar a dar más datos en nuestra evaluación, enfocándose en la calidad semántica de los embeddings generados:

- **Similitud del coseno (Cosine Similarity) [35]:** es una medida ampliamente usada para evaluar la proximidad semántica entre embeddings. Cuanto más alta sea la similitud del coseno entre embeddings de nodos relacionados, más eficazmente estarán capturando estos embeddings las relaciones semánticas reales.
- **Visualización mediante t-SNE [36] o UMAP [37]:** Aunque no es una métrica cuantitativa estricta, es también muy útil utilizar visualizaciones empleando técnicas como t-SNE (t-distributed Stochastic Neighbor Embedding) o UMAP (Uniform Manifold Approximation and Projection) ya que es otra forma efectiva de validar la calidad semántica de los embeddings generados. Estas visualizaciones pueden ofrecer una visión cualitativa sobre la calidad de los embeddings generados ya que se puede observar de forma fácil si los clusters semánticos están bien definidos y son coherentes. Y así reforzar el valor práctico y semántico de los embeddings obtenidos. Estas técnicas funcionan guardando de forma local la estructura semántica de los embeddings originales. Por ello, una visualización se puede considerar que es exitosa si muestra agrupaciones (clusters) claras y coherentes, donde los nodos que están relacionados desde un punto de vista biomédico se van a encontrar agrupados en las mismas zonas del espacio. En ha decidido utilizar las visualizaciones con UMAP en vez de con t-SNE, por su capacidad superior de guardar estructuras globales y locales simultáneamente, lo que puede revelar relaciones más profundas y complejas en los datos biomédicos que se han generado con Bert y BioBERT.

En resumen, el uso combinado de estas métricas y técnicas de evaluación permitirá asegurar de manera robusta y objetiva que los embeddings generados con BERT y BioBERT sean adecuados y beneficiosos para poder entrenar nuestra red de grafos de manera eficiente y cumplir el objetivo central de este trabajo. A continuación, se analizarán de forma detallada y específica los resultados concretos obtenidos con estas métricas sobre los embeddings generados.

4.1.1 Comparación de resultados de algoritmos LLM empleados

En esta sección vamos a comparar los resultados obtenidos de las métricas que se han explicado anteriormente en los modelos BERT utilizados. Los embeddings se han generado siguiendo la misma estrategia: se utilizó un texto específico por nodo compuesto por varios atributos concatenados, y se aplicó *mean pooling* sobre la salida del modelo. La evaluación de la calidad semántica de estos vectores se ha realizado a través de métricas numéricas y visualización en un espacio bidimensional mediante UMAP.

Es importante mencionar que, para la evaluación, se han considerado dos tipos de pares: los **pares positivos** y los **pares negativos**. Los **pares positivos** son aquellos pares de nodos que están relacionados en el grafo, como, por ejemplo, un medicamento y una enfermedad asociada a él. Por otro lado, los **pares negativos** son aquellos pares de nodos que no están directamente relacionados en el grafo. Los pares positivos se utilizan para evaluar cómo de bien el modelo captura las relaciones semánticas entre nodos relacionados, mientras que los pares negativos se utilizan para comprobar si el modelo asigna una baja similitud a los nodos que no deberían estar relacionados. Es decir, los pares negativos ayudan a verificar que el modelo no asigna alta similitud a nodos que no tienen ninguna conexión real, asegurando que las representaciones de los nodos son coherentes con las relaciones del grafo.

Métrica	BERT	BioBERT
Correlación Spearman	0.3768 (p-valor < 0,00001)	0.3929 (p-valor < 0,00001)
Similitud promedio en pares positivos (reales)	0.7138 $\pm$ 0.0737	0.8689 $\pm$ 0.0434
Similitud promedio en pares negativos (aleatorios)	0.5905 $\pm$ 0.1684	0.8176 $\pm$ 0.0704
Diferencia media (positivos - negativos)	0.1233	0.0513

Tabla 4.1: Comparativa de métricas BERT vs BioBERT

Al evaluar numéricamente los embeddings, como se observa en la tabla 4.1, en los generados mediante BioBERT se obtuvo una correlación positiva moderada (Spearman: 0,3929, p-valor < 0,001). Esto indica que los embeddings son capaces de reflejar, al menos en parte, las relaciones biomédicas reales. Aunque el valor de la correlación no alcanza niveles considerados altos (por ejemplo, en torno a 0,7), una correlación cercana a 0.4 puede considerarse muy razonable en un contexto tan complejo como el biomédico. Esto es especialmente relevante ya que hay que tener en cuenta que los textos empleados para generar los embeddings eran relativamente breves y, por tanto, no recogían todo el contexto posible. El p-valor prácticamente nulo respalda la validez estadística del resultado, descartando que la correlación se deba al azar.

En el caso de BERT, que no está preentrenado sobre textos biomédicos, también se ha obtenido una correlación positiva y significativa (Spearman: 0,3768, p-valor < 0,0001). Esto indica que, aunque BERT no es específico del dominio biomédico, con una buena preparación de textos y una estrategia de extracción adecuada (mean pooling), puede conseguir generar re-

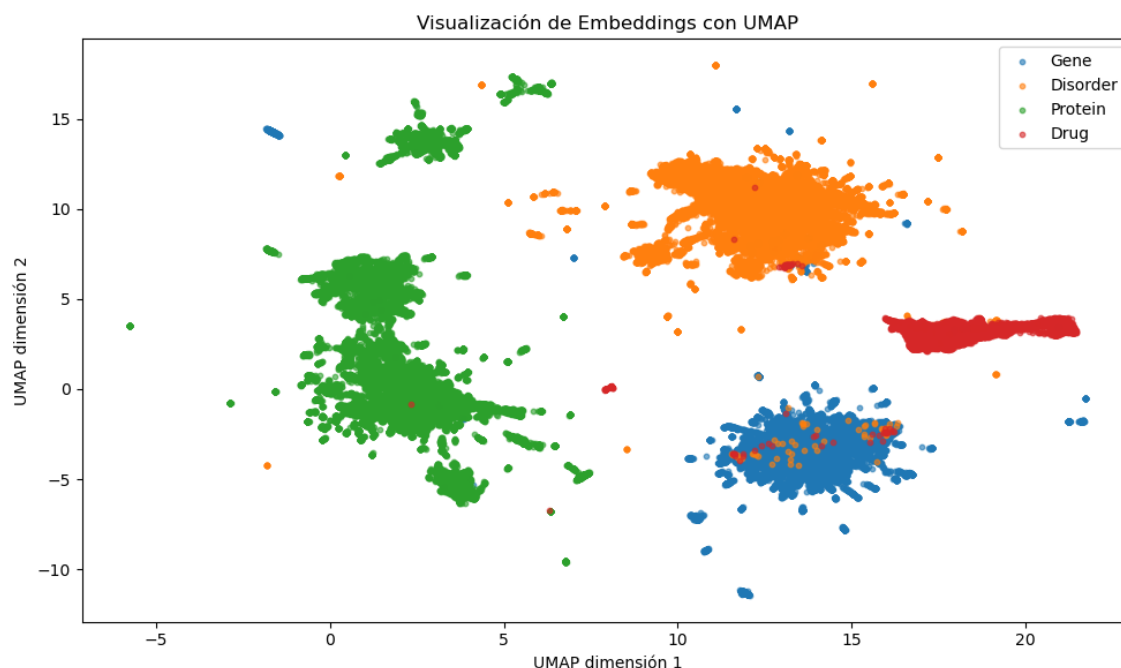


Figura 4.1: Visualización de los embeddings generados con BERT mediante UMAP

presentaciones muy válidas. No obstante, el valor de la correlación es algo inferior al obtenido con BioBERT, lo cual es coherente dada su falta de especialización en el contexto biomédico, pero no hay una gran diferencia significativa entre ambos.

Para comprender mejor los resultados obtenidos y cómo se han evaluado las métricas, se presentan ejemplos específicos de cada tipo de par para ilustrar cómo el modelo maneja las relaciones entre nodos relacionados y no relacionados:

En el caso de los **pares positivos**, consideremos los nodos de tipo **Drug** y **Disorder**. Si tenemos un nodo de medicamento como “Dolasetron” y un nodo de enfermedad como “Hyperaldosteronism”, estos dos nodos están relacionados en el grafo mediante una relación de tipo “DrugHasIndication”. Otro ejemplo de **pares positivos** puede ser el nodo de tipo **Gene**, como “ALDOA” (gene) y un nodo de tipo **Disorder**, como “Azoospermia”, que están relacionados a través de la relación “GeneAssociatedWithDisorder”. Estos nodos deberían tener una alta similitud porque están directamente relacionados. Al aplicar los embeddings generados por BioBERT y BERT, se espera que la similitud entre estos dos nodos sea alta, lo que refleja que ambos están semánticamente relacionados en el dominio biomédico.

Por otro lado, en el caso de los **pares negativos** serían, por ejemplo, un par de nodos como “Dolasetron” (medicamento) y “Bladder calculus” (enfermedad). Aunque ambos son nodos relacionados con el ámbito biomédico, no deberían estar directamente relacionados entre sí

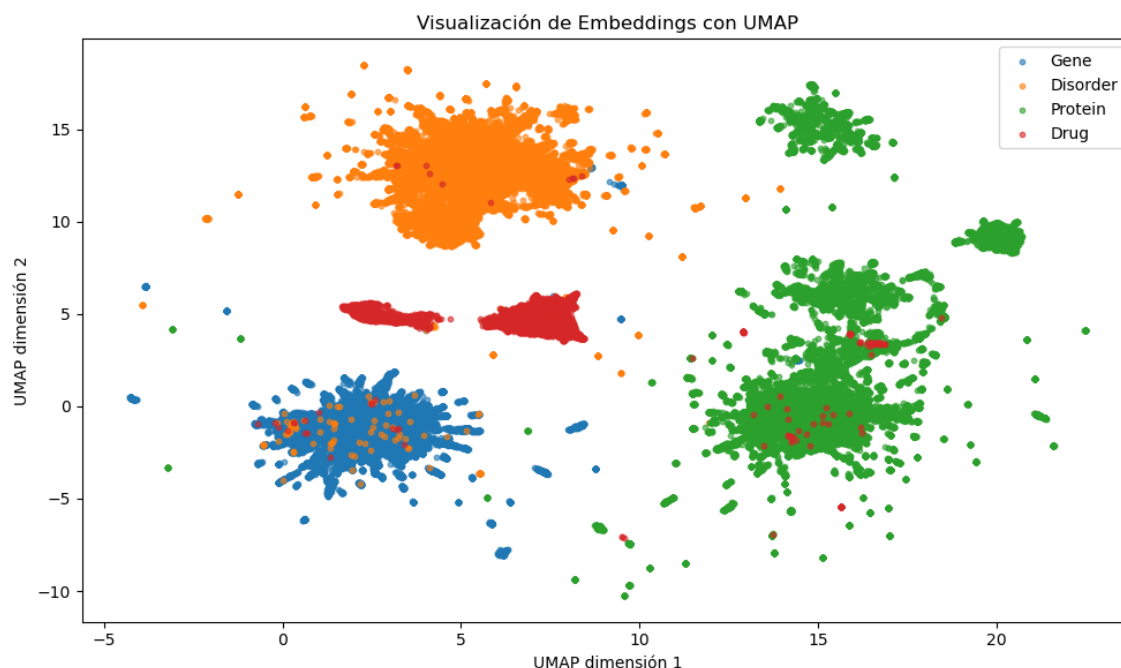


Figura 4.2: Visualización de los embeddings generados con BioBERT mediante UMAP

en el grafo. Otro ejemplo de par negativo podría ser “Dolasetron” (medicamento) y “TP53” (proteína) que no comparten una relación directa en el grafo. Al no estar conectados por una relación explícita, este par también sería considerado negativo. En estos casos, el modelo debería asignar una baja similitud entre estos dos nodos, indicando que no comparten una conexión semántica fuerte.

En cuanto a la métrica de similitud del coseno (4.1), ambos modelos muestran que los nodos realmente relacionados presentan mayor similitud que los pares negativos aleatorios. En el caso de BioBERT, la similitud media para pares positivos es de 0.8689 ± 0.0434 , frente a 0.8176 ± 0.0704 en los negativos, con una diferencia media de 0.0513. En BERT, los pares positivos presentaron una similitud de 0.7138 ± 0.0737 y los negativos de 0.5905 ± 0.1684 , con una diferencia más acusada: 0.1233. Esto sugiere que BERT genera un mayor contraste entre nodos relacionados y no relacionados, aunque con mayor dispersión, lo que puede traducirse en menor estabilidad en los resultados.

Aunque se puede observar que el valor absoluto de la similitud entre pares positivos es mayor en BioBERT, el contraste relativo (positivos vs. negativos) es más claro en BERT. No obstante, la menor desviación estándar de BioBERT en los pares reales (± 0.0434) indica que las relaciones relevantes están representadas de forma más consistente, lo que es clave en nuestra tarea de predicción de enlaces.

Para complementar la evaluación numérica, se ha realizado una visualización mediante UMAP de los embeddings generados por ambos modelos, como se puede observar en las figuras 4.1 y 4.2. En ambos casos se aprecian agrupaciones claras por tipo de nodo. BioBERT ofrece clusters compactos y bien definidos, especialmente en los casos de genes, proteínas y enfermedades, lo que refuerza su coherencia semántica. En el caso de BERT, la estructura general también es reconocible, pero las agrupaciones aparecen algo más dispersas, con mayor mezcla entre categorías, especialmente en los nodos de tipo Drug.

En ambos modelos se observa que los nodos correspondientes a Drug tienden a situarse en posiciones intermedias entre proteínas y enfermedades, lo cual tiene sentido desde el punto de vista biomédico. Los medicamentos suelen actuar como puente entre un mecanismo molecular (representado por las proteínas) y una indicación clínica (la enfermedad). Este patrón visual, especialmente más claro en BioBERT, sugiere que los embeddings captan adecuadamente la semántica relacional del grafo original.

Como conclusión comparativa entre BERT y BioBERT se pueden extraer las siguientes observaciones después de comparar estos modelos.

- **BioBERT** muestra una mejor agrupación visual, especialmente en la organización de los nodos de tipo *Drug*, que tienden a ocupar zonas intermedias entre proteínas y desórdenes, reflejando correctamente su función clínica. Además, su similitud promedio en pares positivos es superior, lo que indica una mayor coherencia semántica.
- **BERT**, aunque no está preentrenado específicamente en texto biomédico, consigue resultados razonables. Muestra una diferencia más acusada entre positivos y negativos en la métrica de similitud del coseno, lo cual puede ser útil para tareas de contraste. Sin embargo, presenta mayor variabilidad en los embeddings de nodos no relacionados.

Por tanto, teniendo en cuenta el objetivo del proyecto, se concluye que **BioBERT es la opción más adecuada** para generar los embeddings que después utilizaremos para enriquecer los datos de entrenamiento de nuestra red. El preentrenamiento sobre textos biomédicos nos ofrece una ventaja en cuanto a consistencia, agrupación semántica y representación de relaciones específicas del dominio biomédico, frente a BERT. No obstante, BERT demuestra ser una alternativa válida cuando se parte de textos bien estructurados, especialmente si se requiere un mayor contraste entre enlaces relacionados y no relacionados.

4.2 Métricas de evaluación para redes neuronales de grafos

A la hora de evaluar modelos de redes neuronales de grafos (GNN) teniendo en cuenta nuestro objetivo de predicción de enlaces (Link Prediction), es fundamental seleccionar métricas adecuadas que permitan valorar la calidad y precisión de las predicciones realizadas por el

modelo [38]. Estas métricas deben reflejar tanto la capacidad del modelo para identificar correctamente enlaces reales (existentes) como su habilidad para priorizar enlaces relevantes en situaciones reales, en nuestro caso sobre el enlace de “DrugHasIndication” entre los nodos y las enfermedades. Para ello, las métricas que se han decidido utilizar, y con las que mediremos los resultados que hemos obtenido, son las siguientes:

- **Área Bajo la Curva ROC (AUC - Area Under the ROC Curve):** Es una métrica muy utilizada para medir la capacidad general que un modelo tiene para diferenciar entre enlaces existentes (positivos) y no existentes (negativos). La AUC se basa en la curva ROC [39], que representa la relación entre la tasa de verdaderos positivos (TPR) y la tasa de falsos positivos (FPR) para diferentes umbrales de clasificación. Un valor de AUC cercano a 1 indica que el modelo tiene una gran capacidad de diferenciación, mientras que un valor cercano a 0.5 indica un rendimiento similar al azar.
- **Accuracy (Exactitud):** Esta métrica mide la proporción de predicciones correctas del modelo en relación con el total de predicciones realizadas. En el contexto de predicciones de enlaces, se calcula como el número de enlaces correctamente clasificados. Es útil para obtener una visión general de la precisión del modelo. Un valor alto de Accuracy indica que el modelo está haciendo una clasificación correcta en la mayoría de los casos, pero no proporciona información detallada sobre el orden de las predicciones, lo cual es crítico en tareas de predicción de enlaces con grandes volúmenes de datos.
- **Recall@K local (Recall en el Top-K):** Esta métrica mide, para cada nodo de tipo *Drug* (o *Disorder*), la proporción de sus enlaces reales *DrugHasIndication* que aparecen entre sus *K* predicciones con mayor score, y luego promedia este valor sobre todos los nodos. Refleja qué porcentaje de todos los enlaces verdaderos el modelo logra recuperar dentro de sus primeras *K* recomendaciones, es decir, su capacidad para no omitir indicaciones reales [40]. Un valor alto de Recall@K local indica que la mayoría de las relaciones verdaderas se encuentran dentro del top-K, lo cual es esencial para garantizar que el modelo sugiere correctamente las relaciones.
- **Precision@K local (Precisión en el Top-K):** Esta métrica calcula, para cada medicamento (o cada enfermedad), qué porcentaje de sus *K* enlaces predichos con mayor puntuación son verdaderos [41]. Después promedia ese porcentaje sobre todos los medicamentos (o enfermedades). Un valor alto de Precision@K local indica que, en el conjunto de las *K* principales recomendaciones, la mayoría son enlaces *DrugHasIndication* correctos, reduciendo los falsos positivos.
- **HitRate@K local (Tasa de acierto en el Top-K):** Esta métrica mide, para cada nodo de tipo *Drug* (o *Disorder*), si al menos una de sus *K* predicciones del enlace *DrugHasIndication* es correcta. Se asigna 1 si hay al menos un acierto, 0 en caso contrario, y promedia este indicador binario sobre todos los nodos. Sirve para estimar la probabilidad de que el modelo incluya al menos una indicación válida dentro de sus *K*

primeras recomendaciones [42]. Un HitRate@K local alto asegura que rara vez se quede sin proponer ninguna relación real en el top-K.

- **Recíproco del Rango Medio (MRR - Mean Reciprocal Rank):** El MRR [40] evalúa el desempeño del modelo considerando el orden relativo de las predicciones. Se calcula tomando la posición (rango) en la que aparece el primer enlace correcto en una lista ordenada de predicciones. El MRR es especialmente útil cuando lo que se interesa evaluar es si el modelo coloca al menos un enlace positivo en posiciones altas de sus rankings de predicción. Valores cercanos a 1 indican excelente desempeño en priorizar los enlaces correctos.
- **Ganancia Acumulada Descontada Normalizada (NDCG - Normalized Discounted Cumulative Gain):** Esta métrica [40] mide la calidad del ranking proporcionado por el modelo, ponderando las predicciones correctas según su posición relativa. Los enlaces correctos situados en las posiciones más altas contribuyen más al valor de NDCG que aquellos situados más abajo. Valores cercanos a 1 nos dicen que el modelo está posicionando correctamente los enlaces más relevantes en los primeros lugares.

Para comprender adecuadamente estas métricas, es importante aclarar previamente algunos conceptos básicos relacionados con la predicción binaria:

- **Verdadero Positivo (True Positive, TP):** Un enlace real que el modelo ha predicho correctamente como existente.
- **Falso Positivo (False Positive, FP):** Un enlace que el modelo ha predicho incorrectamente como existente, pero en realidad no existe.
- **Falso Negativo (False Negative, FN):** Un enlace real que el modelo no ha logrado identificar y ha clasificado incorrectamente como no existente.

En problemas de predicción de enlaces, normalmente no se considera el Verdadero Negativo (True Negative, TN) debido al gran número potencial de enlaces no existentes, lo que podría sesgar el análisis. A continuación se mostrarán los resultados que se han obtenido a partir de estas métricas. Y así poder evaluar de forma cuantitativa y objetiva el rendimiento que está teniendo nuestro modelo entrenado.

4.3 Resultados de los modelo GNN implementados

En esta sección se presentan y analizan los resultados obtenidos de las métricas anteriores utilizando distintos modelos de redes neuronales en grafos (GNN), concretamente GraphSAGE y GAT. El objetivo es comparar diversas configuraciones arquitectónicas y combinaciones

de hiperparámetros para identificar qué modelo y estructura ofrecen un mejor rendimiento en la tarea de predicción de enlaces entre medicamentos y enfermedades dentro del grafo heterogéneo.

Cada modelo ha sido entrenado y evaluado sobre las mismas particiones de datos y utilizando los mismos embeddings iniciales, permitiendo así una comparación justa y directa entre sus resultados. A continuación, se detallan las configuraciones exploradas para GraphSAGE y GAT, junto con sus respectivas motivaciones:

Configuraciones exploradas con GraphSAGE:

- **Modelo inicial con reducción de dimensionalidad final:** Esta configuración de GraphSAGE fue la primera que se testeó, se basa en tres capas convolucionales (SAGEConv), con una reducción progresiva hasta alcanzar una dimensión final de 32. El objetivo es obtener embeddings compactos que retengan la información relevante sin incurrir en sobreajuste, ayudados también por técnicas de regularización como el dropout.
- **Modelo profundo con BatchNorm y sin reducción:** En esta versión se quiso mejorar a la anterior y se explora un modelo más profundo, de hasta cuatro capas, sin reducción de la dimensión (manteniéndose en 768). Se incluye Batch Normalization para facilitar un entrenamiento más estable y eficiente. Esta configuración busca conservar al máximo la información de entrada y poder evaluar si hay un impacto en el rendimiento del modelo. Los resultados que se detallarán en el apartado posterior serán con esta configuración ya que es una mejora del modelo inicial que se exploró.
- **Clasificador basado en producto escalar vs. MLP:** Además de las capas de GraphSAGE, se han probado dos enfoques para la etapa final de clasificación de enlaces. Por un lado, el producto escalar directo entre los embeddings de los nodos implicados, ofreciendo una solución sencilla y eficiente en términos computacionales. Por otro, un clasificador MLP que concatena los embeddings y los pasa por capas adicionales. Aunque ambos métodos han sido considerados, los resultados que se presentan en las secciones posteriores corresponden únicamente al enfoque con MLP, ya que ha demostrado ofrecer un mejor rendimiento en las tareas evaluadas.

Configuraciones exploradas con GAT:

- **Modelo GAT básico con atención multi-cabeza:** Esta arquitectura emplea dos capas de atención (GATConv), utilizando múltiples cabezas para capturar diferentes perspectivas de los vecinos de cada nodo. Se parte del mismo embedding inicial que en GraphSAGE y se explora cómo la atención afecta a la representación final de los nodos.

- **Mismo clasificador que en GraphSAGE:** Al igual que en el caso anterior, se han evaluado dos opciones para la capa de salida: el producto escalar entre embeddings, que representa una opción simple y eficiente y un MLP que permite modelar relaciones más complejas gracias a la concatenación de los embeddings y su posterior procesamiento en capas densas adicionales. De esta forma se mantiene la consistencia en la comparación entre arquitecturas. Los resultados que se presentan para el modelo GAT han sido los obtenidos utilizando el clasificador MLP, ya que este enfoque ha mostrado un rendimiento superior en comparación con el método basado en producto escalar.

Para cada una de las configuraciones finales elegidas se ha realizado un análisis detallado, incluyendo:

- Evaluación numérica mediante las métricas establecidas (AUC, Recall@k, Precision@k, HitRate@k MRR, NDCG).
- Evaluación de las curvas de pérdidas en los conjuntos de entrenamiento y validación.
- Evaluación de las predicciones directas sobre el enlace de “DrugHasIndication”.

Estos resultados se presentan a continuación. Con el objetivo de analizarlos para poder determinar cuál es la opción más adecuada y eficiente para alcanzar los objetivos planteados en este proyecto.

4.3.1 Configuraciones y parámetros de entrenamiento

Las configuraciones seleccionadas se recogen a continuación. La idea principal es utilizar configuraciones que mantengan un equilibrio entre la mayor precisión posible y mantener unos tiempos de entrenamiento razonables desde el punto de vista computacional:

4.3.1.1 Configuración 1: Binary Cross-Entropy y GraphSAGE

En esta configuración se va a entrenar el modelo haciendo uso de la función de pérdida Binary Cross-Entropy (BCE) en lugar de Margin Ranking Loss, que se verá más adelante. El objetivo principal es evaluar si una formulación más clásica de clasificación binaria mejora los resultados en la tarea de predicción de enlaces del tipo DrugHasIndication. La arquitectura general del modelo se mantendrá constante para las dos formas de entrenar, para así permitir una comparación más real:

- **Modelo:** GNN heterogéneo con 3 capas (GraphSAGE + BatchNorm + Dropout)
- **Tipo de clasificador:** MLP con concatenación Drug-Disorder

- **Embeddings de entrada:** BioBERT (dimensión 768)
- **Reducción de dimensión:** a 128 mediante proyección lineal
- **Loss:** Binary Cross-Entropy (BCE)
- **Optimización:** AdamW
- **Learning rate:** $1 \cdot 10^{-4}$
- **Weight decay:** $1 \cdot 10^{-4}$
- **Dropout:** 0.3
- **Patience early stopping:** 10 épocas
- **Épocas de entrenamiento:** 100
- **Métricas monitorizadas:** AUC, MRR local, NDCG@50, train/val loss

La arquitectura de la red neuronal está compuesta por tres capas de tipo GraphSAGE, cada una seguida por normalización por lotes (BatchNorm) y activación ReLU. Se incorporó dropout como técnica de regularización. El clasificador final utilizado es un MLP que concatena las representaciones de los nodos Drug y Disorder y predice la probabilidad de existencia del enlace DrugHasIndication.

Se han utilizado embeddings generados con BioBERT para todos los nodos del grafo. Estos embeddings se proyectan a una dimensión menor (128) mediante una proyección lineal (capa de neuronas totalmente conectada, ver arquitectura de la red) antes de ser procesados por el modelo GNN, lo que ayuda a reducir la complejidad computacional sin perder información semántica relevante.

Para el entrenamiento se ha utilizado el optimizador AdamW, junto con una política de early stopping basada en la métrica AUC sobre el conjunto de validación. Durante el entrenamiento se han monitorizado las métricas que se han explicado anteriormente: AUC, MRR y NDCG@50, con el objetivo de evaluar la capacidad del modelo no solo para distinguir entre enlaces positivos y negativos, sino también para ordenar correctamente las predicciones. La Figura 4.3 muestra la evolución de la pérdida de entrenamiento y validación a lo largo de las épocas.

Se puede observar que la función de pérdida BCE disminuye de forma progresiva, mostrando una buena capacidad de generalización del modelo. A partir de la época 60 se aprecia una estabilización en la pérdida de validación, lo que indica una convergencia del aprendizaje. Las métricas que se alcanzan con esta configuración se verán en la sección siguiente.

Este comportamiento sugiere que, aunque la función de pérdida BCE no está pensada específicamente para tareas de ranking, puede ser una opción muy válida en problemas de predicción

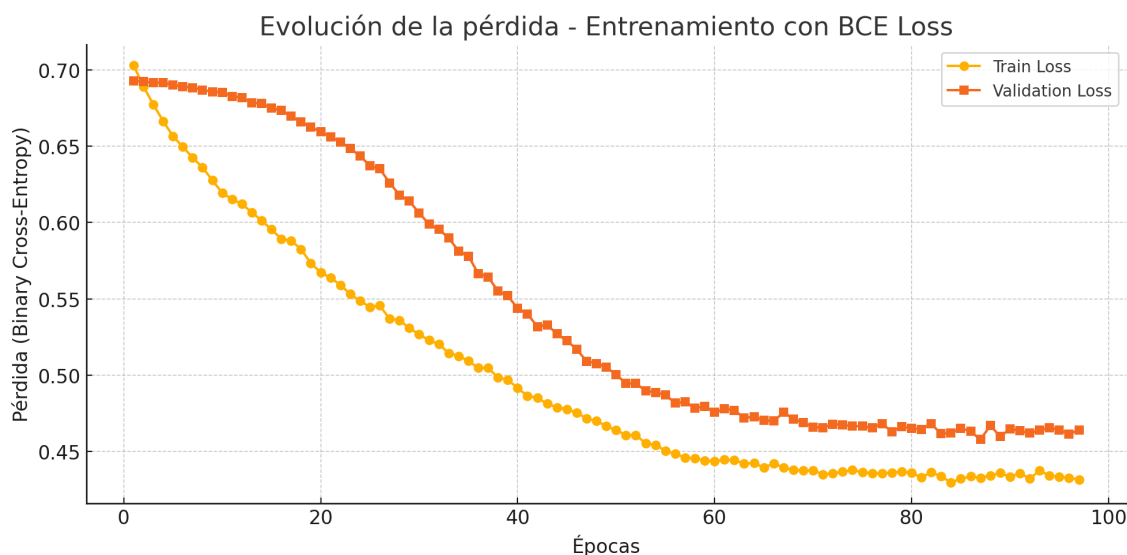


Figura 4.3: Curva de pérdida durante el entrenamiento con BCE y GraphSAGE

de enlaces. Especialmente si se combina con una arquitectura robusta y unos embeddings completos como los que nos aporta BioBERT. Además, el entrenamiento con BCE puede resultar más estable y sencillo de ajustar, por lo que puede ser el enfoque más óptimo a la hora de elegir una configuración.

4.3.1.2 Configuración 2: Margin Ranking Loss y GraphSAGE

Para el entrenamiento del modelo con la pérdida Margin Ranking, los parámetros y especificaciones del modelo han sido las siguientes:

- **Modelo:** GNN heterogéneo con 3 capas (GraphSAGE + BatchNorm + Dropout)
- **Tipo de clasificador:** MLP con concatenación Drug–Disorder
- **Embeddings de entrada:** BioBERT (dimensión 768)
- **Reducción de dimensión:** a 128 mediante proyección lineal
- **Loss:** Margin Ranking Loss (margin = 1.0)
- **Optimización:** AdamW
- **Learning rate:** $1 \cdot 10^{-4}$
- **Weight decay:** $1 \cdot 10^{-4}$

- **Dropout:** 0.3
- **Patience early stopping:** 10 épocas
- **Épocas de entrenamiento:** 100
- **Métricas monitorizadas:** AUC, MRR local, NDCG@50, train/val loss

Los embeddings se generan de la misma forma que antes y la arquitectura de la red neuronal también es la misma que en la anterior configuración. En esta versión se ha optado por utilizar Margin Ranking Loss como función de pérdida, al estar especialmente diseñada para tareas de ranking, por lo que puede ser muy útil en tareas de predicción de enlaces en grafos.

Durante el entrenamiento se han monitorizado las métricas de evaluación global (AUC) y de ranking local por nodo origen (MRR local y NDCG@50), además de las curvas de pérdida. La Figura 4.4 muestra la evolución de la función de pérdida a lo largo de las 100 épocas entrenadas en los conjuntos de entrenamiento y en el de validación.

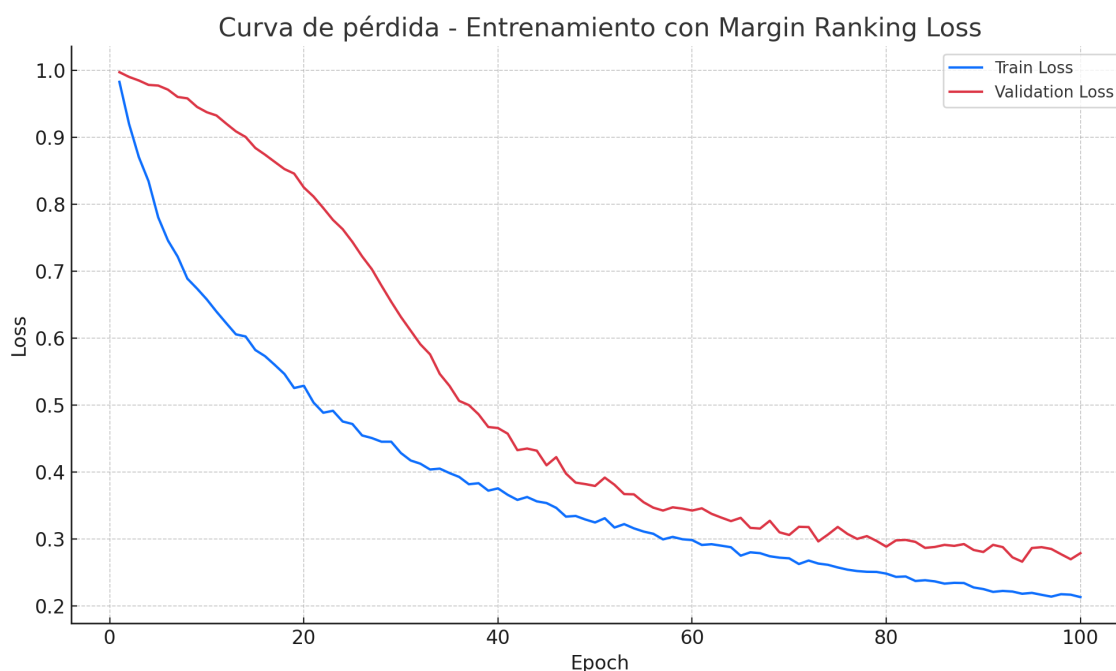


Figura 4.4: Curva de pérdida durante el entrenamiento con MRL y GraphSAGE

Como se puede ver, la función de pérdida va disminuyendo de forma progresiva hasta estabilizarse en torno a la época 90, lo que indica que el modelo ha convergido correctamente y no hay señales de sobreajuste. Las métricas que se alcanzan con esta configuración se presentan en la sección siguiente.

4.3.1.3 Configuración 3: Binary Cross-Entropy y GAT

Con el objetivo de explorar otras arquitecturas de redes neuronales de grafos y comprobar si pueden ofrecer mejoras respecto a GraphSAGE, se ha incorporado una tercera configuración utilizando la arquitectura GAT (Graph Attention Networks). La idea es ver si al introducir mecanismos de atención, que permiten al modelo dar más peso a ciertos vecinos en función de su relevancia, se pueden mejorar los resultados en un grafo tan heterogéneo como el que tenemos. Además, también se quería hacer esta comparación para así poder entender mejor cómo responde cada arquitectura ante la complejidad de las relaciones entre medicamentos y enfermedades.

Para el entrenamiento del modelo con la arquitectura GAT y Binary Cross Entropy Loss (BCE), los parámetros y especificaciones del modelo han sido los siguientes:

- **Modelo:** GNN heterogéneo con 3 capas (GAT + BatchNorm + Dropout)
- **Tipo de clasificador:** MLP con concatenación Drug-Disorder
- **Embeddings de entrada:** BioBERT (dimensión 768)
- **Reducción de dimensión:** a 128 mediante proyección lineal
- **Loss:** Binary Cross Entropy Loss
- **Optimización:** AdamW
- **Learning rate:** $1 \cdot 10^{-4}$
- **Weight decay:** $1 \cdot 10^{-4}$
- **Dropout:** 0.3
- **Patience early stopping:** 10 épocas
- **Épocas de entrenamiento:** 120
- **Métricas monitorizadas:** AUC, MRR local, NDCG@50, train/val loss

En esta configuración, también se han empleado embeddings generados por BioBERT, proyectados a una dimensión menor (128) para reducir la complejidad computacional pero conservando el valor que aportan los embeddings. Estos embeddings fueron procesados por la arquitectura GAT, que utiliza atención para mejorar la capacidad del modelo de aprender representaciones relevantes en un grafo heterogéneo.

El modelo se entrenó utilizando Binary Cross Entropy Loss y AdamW como optimizador, y el entrenamiento fue monitoreado mediante Early Stopping basado en la métrica AUC sobre el conjunto de validación, para prevenir que se genere overfitting.

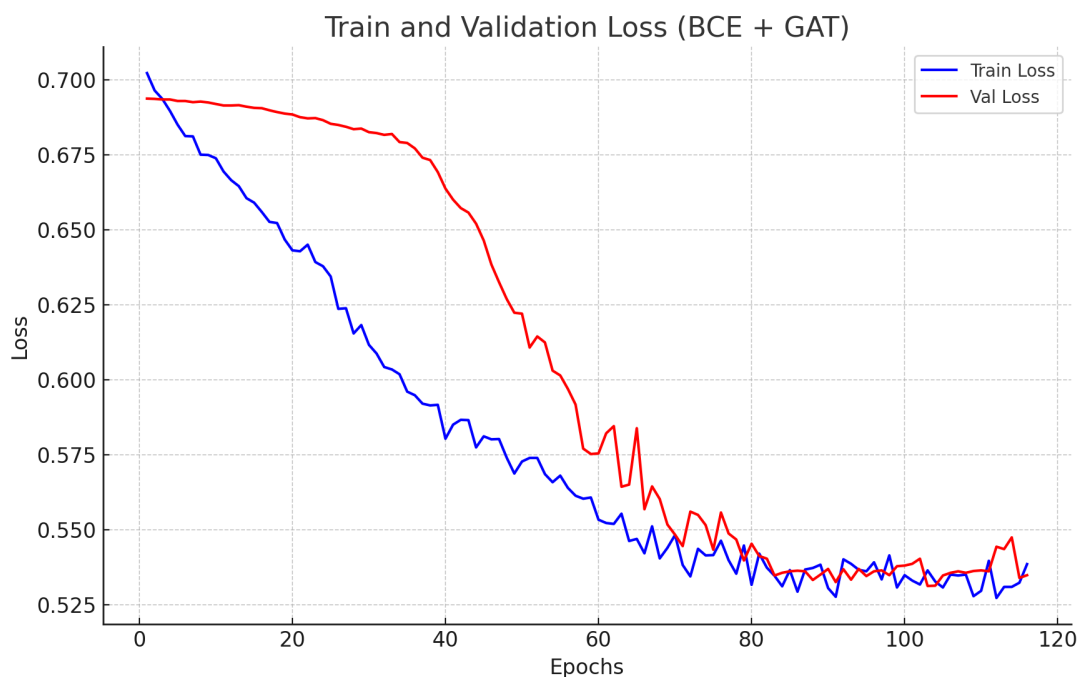


Figura 4.5: Curva de pérdida durante el entrenamiento (Configuración 3)

Durante el entrenamiento, se monitorizaron no solo la pérdida de entrenamiento y validación, sino también las métricas de rendimiento global (AUC) y de ranking local por nodo (MRR local y NDCG@50) como en las otras configuraciones. La Figura 4.5 muestra la evolución de la pérdida durante las 120 épocas entrenadas.

A lo largo del entrenamiento, la pérdida de entrenamiento disminuye de forma constante, estabilizándose en torno a la época 90. En la pérdida de validación también se muestra una tendencia decreciente, lo que indica que el modelo está generalizando bien sin sobreajustarse.

En las siguientes secciones, se presentarán los resultados de las métricas obtenidas para las distintas configuraciones, así como los resultados de las predicciones directas de medicamentos y enfermedades.

4.3.2 Comparación de resultados con diferentes configuraciones y enfoques

En esta sección se comparan los resultados obtenidos de las métricas para las diferentes configuraciones de modelos evaluados. Primero, se presentarán los resultados de las métricas en el conjunto de validación, con el objetivo de identificar qué configuración ofrece el mejor rendimiento sin riesgo de sesgar los resultados. Después, se mostrarán los resultados en el conjunto de test para evaluar el rendimiento final del modelo.

A continuación, se muestra un resumen de los experimentos realizados y las métricas evaluadas:

- **Modelos evaluados:** GraphSAGE con Binary Cross Entropy (BCE), GraphSAGE con Margin Ranking Loss (MRL) y GAT con Binary Cross Entropy (BCE).
- **Métricas evaluadas en validación:** AUC, MRR local y NDCG@50. Con el objetivo de medir la capacidad de discriminación y la calidad del ranking sin fijar un umbral de corte.
- **Métricas evaluadas en test:** AUC, Accuracy, Recall@5, Precision@5, HitRate@5, MRR local y NDCG@50. Para así evaluar el rendimiento general del modelo y la calidad de las recomendaciones bajo un umbral realista.
- **Objetivo:** Comparar el rendimiento de estos modelos en la tarea de predicción del enlace “DrugHasIndication”, entre los nodos de tipo *Drug* y *Disorder*.

A continuación, en la tabla 4.2, se muestran los resultados de las métricas evaluadas para los modelos en el conjunto de validación, que incluyen: AUC, MRR Local y NDCG@50.

Métrica	GraphSAGE (BCE)	GraphSAGE (MRL)	GAT (BCE)
AUC	0.8721	0.8930	0.8341
MRR Local	0.9631	0.9523	0.9620
NDCG@50	0.9641	0.9732	0.9712

Tabla 4.2: Comparativa de métricas en el conjunto de validación

Analizando los resultados obtenidos en el conjunto de validación, presentados en la tabla 4.2, se puede ver que el modelo GraphSAGE con Margin Ranking Loss (MRL) destaca con el mejor rendimiento, especialmente en la métrica AUC con un valor de **0.8930**, lo que indica que tiene una gran capacidad para diferenciar entre los pares de nodos relacionados y los no relacionados. Además, el modelo GraphSAGE con MRL también muestra un rendimiento destacado en las métricas de **MRR Local** y **NDCG@50** (0.9523 y 0.9732, respectivamente), lo que confirma su buena capacidad para priorizar correctamente las relaciones que son relevantes.

Por otro lado, GraphSAGE con BCE también obtiene muy buenos resultados, con un AUC de **0.8721** y un MRR Local de **0.9631**, aunque no alcanza el nivel de MRL en AUC. Por otro lado, la configuración con GAT presenta un rendimiento algo inferior, con un **AUC de 0.8341**. Esto sugiere que los mecanismos de atención de GAT podrían no ser tan efectivos para capturar las relaciones en esta tarea de predicción de enlaces en comparación con los enfoques anteriores.

A continuación, en la tabla 4.3, se muestran los resultados de las métricas evaluadas para los modelos en el conjunto de test:

Métrica	GraphSAGE (BCE)	GraphSAGE (MRL)	GAT (BCE)
AUC	0.8685	0.8688	0.8235
Accuracy	0.7960	0.7836	0.7503
Recall@5 Local	0.9371	0.9438	0.9440
Precision@5	0.6688	0.7323	0.4082
HitRate@5	0.9367	0.9813	0.9993
MRR Local	0.8381	0.9313	0.8985
NDCG@50	0.8749	0.9447	0.9225

Tabla 4.3: Comparativa de métricas en el conjunto de test

Al analizar los resultados de las métricas en el conjunto de test, presentados en la tabla 4.3, se observa que todos los modelos presentan un AUC superior a 0.8 y una Accuracy superior a 0.75, lo que indica una buena capacidad para diferenciar entre enlaces reales y negativos. Sin embargo los modelos que usan GraphSAGE obtienen valores en estas métricas superiores a la configuración introduciendo GAT. Destacando el modelo GraphSAGE con Margin Ranking Loss (MRL), con un AUC de 0.8688 y el modelo GraphSAGE con Binary Cross Entropy (BCE), con una Accuracy de 0.7960.

En cuanto a la métrica de Recall@5, todos los modelos muestran un valor superior a 0,9, lo que significa que casi todas las indicaciones verdaderas aparecen dentro de las cinco primeras predicciones. En esta métrica la configuración con la que se obtiene el mejor resultado es con el modelo GAT (BCE), pero con muy poca variación respecto a los otros dos.

En la métrica de Precisión@5 podemos ver la mayor diferencia de resultados entre los tres modelos. Para la configuración de GAT (BCE) se obtiene el valor más bajo (0.4082), lo que significa que menos de la mitad de sus cinco primeras recomendaciones son aciertos verdaderos. GraphSAGE con BCE mejora significativamente hasta 0.6688, y GraphSAGE con MRL alcanza el valor más alto (0.7323), es decir, casi siete de cada diez predicciones en su top-5 son enlaces correctos.

Analizando la métrica de HitRate@5. se puede observar que todos los modelos tienen un valor muy alto, cercano a 1.0 . Esto implica que casi todos los nodos reciben al menos un enlace verdadero en su top-5. En nuestros resultados, GAT (BCE) alcanza 0.9993, GraphSAGE con MRL 0.9813 y GraphSAGE con BCE 0.9367, lo que confirma que para prácticamente todos los nodos el modelo propone al menos una relación válida dentro de sus cinco sugerencias principales.

Por último, en cuanto a las métricas de MRR Local y NDCG@50, el modelo que obtiene mejores resultados es la configuración de GraphSAGE con MRL, con MRR Local (0.9313) y NDCG@50 (0.9447), lo que refleja una capacidad superior para ordenar correctamente las predicciones y dar más relevancia a las relaciones correctas. Por otro lado, el modelo GAT, en estas métricas también tiene un muy buen rendimiento, especialmente en **NDCG@50** (0.9231).

En resumen, el uso de GraphSAGE mejora el rendimiento en comparación con GAT, tanto en el conjunto de validación como en el de test. Dentro de los modelos GraphSAGE, tanto BCE como MRL ofrecen un rendimiento similar, con MRL mostrando una ligera ventaja en las métricas clave, especialmente en el conjunto de test. Se puede afirmar que configuraciones de GraphSAGE son competitivas y viables. En resumen, GraphSAGE es claramente superior a GAT, y la elección entre BCE y MRL depende de si se quiere priorizar relaciones más relevantes o un enfoque más general.

4.4 Predicción enlace DrugHasIndication

Tras evaluar el rendimiento de los modelos a través de las anteriores métricas, nos falta analizar la capacidad del sistema para predecir directamente la relación DrugHasIndication, es decir, si el modelo es capaz de identificar medicamentos relevantes para una enfermedad dada, y viceversa, si es capaz de identificar enfermedades asociadas a un medicamento dado. Además, se analizará su capacidad para identificar posibles asociaciones drug-disorder no conocidas previamente. Esta tarea, conocida como reposicionamiento farmacológico (drug repurposing), tiene una gran utilidad cuando se trata, sobre todo, de enfermedades raras o sin tratamientos eficaces. Estos son los objetivos principales del proyecto y para lo que se han entrenado los modelos.

Esta estrategia permite una evaluación más cualitativa del modelo, ya que se pueden analizar casos concretos para comprobar si las predicciones resultan coherentes y clínicamente viables. Además, se analizarán tanto casos con predicciones más acertadas como aquellos donde el modelo falla o devuelve scores bajos, lo cual aporta información relevante sobre sus limitaciones y posibles mejoras.

Nota: Para interpretar los resultados desde el punto de vista clínico, se ha consultado información sobre los fármacos y enfermedades mencionadas en fuentes médicas de referencia como PubChem¹, DrugBank² y MedlinePlus³, además de artículos científicos disponibles públicamente. Estas fuentes han servido de apoyo para valorar si las asociaciones predichas por el modelo tienen sentido desde un punto de vista farmacológico o si podrían representar posibles casos de reposicionamiento.

4.4.1 GraphSAGE y Binary Cross Entropy

En primer lugar, se van a analizar los resultados obtenidos con el modelo entrenado utilizando Binary Cross Entropy (BCE) como función de pérdida. Este modelo emplea una arquitectura GNN y un clasificador MLP, utilizando embeddings iniciales generados por BioBERT, los

¹<https://pubchem.ncbi.nlm.nih.gov>

²<https://go.drugbank.com>

³<https://medlineplus.gov>

cuales son luego proyectados a una dimensión de 128. El objetivo principal de este modelo es la predicción de relaciones entre nodos del tipo Drug y Disorder, es decir, predecir qué medicamentos están asociados a qué enfermedades.

El funcionamiento de este modelo consiste en generar un embedding vectorial para cada nodo del grafo, capturando tanto las características estructurales como semánticas de cada nodo. A continuación, el clasificador MLP toma como entrada los embeddings de un par de nodos (uno del tipo Drug y otro del tipo Disorder) y calcula una puntuación que indica la probabilidad de que exista una relación entre ellos, utilizando Binary Cross Entropy. Esta puntuación se obtiene comparando los resultados predichos con las etiquetas reales (positivas o negativas), es decir, el modelo trata de predecir si una relación entre un fármaco y una enfermedad existe o no.

Para todos los posibles pares de nodos entre el tipo Drug y Disorder, el modelo genera una probabilidad de relación, ordenando estos pares según su score descendente. Este enfoque se utiliza para obtener un ranking de las enfermedades más asociadas a un fármaco, o de los fármacos más relevantes para una enfermedad concreta, basándose en las probabilidades de las relaciones.

Resultados desde Disorder → Drugs

Se ha explorado el comportamiento del modelo al predecir los cinco medicamentos con mayor score para distintas enfermedades. Es importante señalar que, en este análisis, no se han considerado pares de nodos que ya estén explícitamente conectados en el grafo (es decir, aquellos pares de nodos que ya forman parte de las relaciones existentes entre los nodos). Por ejemplo, si “Diabetes” e “Insulina” están conectados en el grafo, no se consideran como parte de las predicciones, ya que esa relación ya está conocida en el grafo y no representa una nueva predicción del modelo. Las predicciones se harán con nodos no vistos en el entrenamiento. La tabla 4.4 muestra los resultados obtenidos para cinco enfermedades seleccionadas, cuyas predicciones se analizan a continuación.

Para la primera enfermedad que vamos a analizar, el Alzheimer, el modelo recomienda en primer lugar varios inhibidores de la ECA: Benazeprilat, Moexiprilat y Ramiprilat, junto a Brofaromine, un inhibidor de la MAO-A, y Endralazine, que también tiene actividad vasodilatadora. Todos ellos cuentan con estudios que apuntan a efectos neuroprotectores y a la mejora del flujo sanguíneo cerebral, dos mecanismos clave en el Alzheimer. En otras palabras, la red ha identificado fármacos con un perfil terapéutico coherente con la fisiopatología de la enfermedad, lo que refuerza la idea de que está captando patrones biomédicos reales y sugiere oportunidades para posibles estudios de reposicionamiento farmacológico.

Para la segunda enfermedad, carcinoma maligno de colon, el modelo destaca en primer lugar Albutrepenonacog alfa, un factor de coagulación útil para controlar sangrados tumorales. Le siguen Dexverapamil, un derivado de un medicamento cardiovascular que estudios preliminares

Enfermedad	Score Promedio	Top-5 Fármacos
Alzheimer	0.8761	Benazeprilat (0.9103) Brofaromine (0.8994) Endralazine (0.8745) Moexiprilat (0.8484) Ramiprilat (0.8480)
Malignant colon neoplasm	0.9662	Albutrepenonacog alfa (0.9752) Dexverapamil (0.9656) Desmethylsertraline (0.9653) MK-0752 (0.9630) Darunavir (0.9620)
Depressive disorder	0.9701	Gentamicin (0.9744) TAS-116 (0.9713) Ibalizumab (0.9704) Mosapramine (0.9676) Lumateperone (0.9670)
Lung disorder	0.8589	Spesolimab (0.8839) Soraphen A (0.8642) Ramucirumab (0.8505) Tetraferic tricitrate (0.8494) S-L-C ^a (0.8467)

Tabla 4.4: Top 5 medicamentos para el modelo GraphSAGE con Binary Cross Entropy

^a *S-(Methylmercury)-L-Cysteine*

sugieren que podría interferir con el crecimiento de células cancerosas. Desmethylsertraline, un metabolito de un antidepresivo, también ha mostrado en estudios en laboratorios capacidad para frenar la proliferación celular. MK-0752 es un compuesto en fase de ensayo clínico contra varios tumores, investigado en cáncer de colon. Y por último, el modelo sugiere Darunavir, un antiviral, que ha revelado actividad antiproliferativa en cultivos celulares. En conjunto, estos resultados combinan tratamientos consolidados con candidatos más novedosos, confirmando que el modelo es capaz de descubrir tanto usos alternativos de fármacos conocidos como posibilidades de reposicionamiento.

Para la depresión, el modelo acierta plenamente al situar en el top, medicamentos que tienen clara actividad antidepresiva como el lumateperone y mosapramine, lo que demuestra que ha captado bien las indicaciones clínicas. Pero a la vez, aparecen compuestos menos esperados como Gentamicin, TAS-116 e Ibalizumab, que no se usan habitualmente en salud mental. Estas opciones podrían revelar nuevas vías de acción, ligadas a la modulación del sistema inmune o a señales metabólicas, y merecen investigarse como posibles enfoques de reposicionamiento.

Para enfermedad pulmonar, lung disorder, el modelo destaca principalmente fármacos que tienen actividad antiinflamatoria y antitumoral. En primer lugar aparece Spesolimab, un anticuerpo contra el receptor de IL-36 que se está explorando por su capacidad de calmar la inflamación pulmonar. Soraphen A, que actúa sobre rutas metabólicas clave implicadas en la respuesta inmune. También aparece en el top Ramucirumab, un anticuerpo anti-VEGFR2 aprobado en cáncer de pulmón que refuerza la idea de un componente vascular en estas patologías. Además, el modelo indica el medicamento Tetraferic tricitrato, un complejo de hierro que podría mejorar el transporte de oxígeno en pacientes con defectos en la captación de hierro. El último fármaco, S-(Methylmercury)-L-Cysteine, apunta más bien a ruido de datos o efectos colaterales, ya que a priori este compuesto no está relacionado con enfermedades pulmonares. El modelo capta tanto tratamientos consolidados como candidatos novedosos a explorar.

Algunas de las conclusiones que podemos sacar respecto a los resultados con esta configuración son los siguientes:

- Organización semántica eficaz: El modelo sitúa en los primeros puestos fármacos con mecanismos plausibles (por ejemplo IECA en Alzheimer o antivirales en cáncer de colon), lo que demuestra que ha aprendido correctamente la estructura biomédica del grafo.
- Impulso a nuevas hipótesis terapéuticas: El modelo ha generado altas puntuaciones a compuestos poco convencionales (anticuerpos anti-IL en trastornos pulmonares, antibióticos o inmunomoduladores en depresión), revelando candidatos atractivos para futuros estudios de reposicionamiento.

Resultados desde Drugs → Disorder

A continuación, se explora la capacidad del modelo para identificar las enfermedades más relevantes dado un medicamento concreto. En la tabla 4.5, se muestran los cinco fármacos con sus cinco enfermedades predichas con mayor puntuación. Esta perspectiva inversa es especialmente útil para estudiar el potencial de cada fármaco en el contexto del *drug repositioning*, detectando nuevas posibles indicaciones o relaciones no exploradas. Se van a analizar los resultados obtenidos desde un punto de vista clínico y farmacológico.

Para Cetuximab, el modelo acierta al identificar efectos cutáneos y hormonales, predice con alta puntuación queratosis pilaris atrófica y anovulación, ambas vinculadas a su acción sobre EGFR en piel y ovario. Además, sugiere potencial en tumores blandos como el histiocitoma angiomatoides fibroso (donde EGFR puede desempeñar un papel), lo que concuerda con su uso oncológico. Las predicciones relativas a miopía y atrofia óptica son menos evidentes clínicamente y probablemente reflejan conexiones de red más que indicaciones reales. Estos resultados muestran que el modelo captura bien tanto los efectos secundarios esperables como posibles nuevas aplicaciones tumorales.

Fármaco	Score Promedio	Top-5 Enfermedades
Cetuximab	0.9750	Histiocytoma, Angiomatoid (0.9222) Anovulation (0.9889) Keratosis pilaris atrophicans (0.9883) Myopia 26 (0.9880) Optic atrophy (0.9877)
Denileukin diftitox	0.9869	Multiple sclerosis (0.9882) Lipoatrophic diabetes (0.9873) Immunodeficiency 51 (0.9867) Hutchinson–Gilford progeria (0.9866) 4q11.2 microduplication (0.9857)
Etanercept	0.9684	Gonadal agenesis (0.9804) Graham LPL ^a (0.9800) Schizophrenia 12 (0.9346) Atransferrinemia (0.9746) Acetyl-CoA carboxylase (0.9728)
Sermorelin	0.9686	Chronic atrophic gastritis (0.9755) Anovulation (0.9728) Severe hemophilia A (0.9670) Keratitis (0.9652) Noonan syndrome 5 (0.9628)

Tabla 4.5: Top 5 enfermedades para el modelo GraphSAGE con Binary Cross Entropy

^a *Graham Little–Piccardi–Lassueur*

Para Denileukin diftitox, el modelo acierta al priorizar trastornos con claro componente inmunológico, como la esclerosis múltiple y la inmunodeficiencia tipo 51, lo cual encaja con su acción sobre células T. También sugiere diabetes lipoatrófica, una enfermedad con base autoinmune. Sin embargo, predice el síndrome de Hutchinson–Gilford y la microduplicación 14q11.2, que carecen de respaldo clínico evidente, por lo que probablemente reflejen conexiones de red más que indicaciones terapéuticas reales.

En el caso de Etanercept, el modelo acierta al proponer patologías de base inflamatoria o autoinmune, como la gonadal agenesis y el síndrome de Graham Little–Piccardi–Lassueur—coherentes con su acción anti-TNF. Otras asociaciones, como “schizophrenia 12” o atransferrinemia, resultan menos plausibles y reflejan probablemente patrones de conectividad del grafo. En conjunto, predice correctamente el componente inflamatorio del fármaco, pero sugiere vínculos que deberían confirmarse con criterio clínico de expertos.

En cuanto a Sermorelin, tiene sentido que el modelo lo relacione con Noonan syndrome 5 y anovulación, ya que ambos pueden mejorar al estimular la secreción de la hormona del crecimiento (GH). También cuadra la asociación con queratitis, dada su posible acción en la

reparación ocular. En cambio, la gastritis atrófica crónica y la hemofilia A grave no guardan ninguna relación directa a simple vista con el mecanismo de GH.

En resumen, desde un punto de vista clínico, el modelo demuestra una muy buena capacidad para asociar fármacos y enfermedades de forma coherente con mecanismos conocidos. Además el modelo es capaz de identificar no solo indicaciones potenciales, sino también predicciones menos evidentes que reflejan la riqueza semántica del grafo y pueden servir de punto de partida para estudios de reposicionamiento. Hay que tener en cuenta que las predicciones deberían revisarse con especialistas, ya que algunas asociaciones pueden responder más a la proximidad en el grafo que a indicaciones reales, por lo que deberían revisarse antes de sacar conclusiones, ya que pueden apuntar a hipótesis novedosas que merecen un análisis bibliográfico o experimental previo.

4.4.2 GraphSAGE y Margin Ranking Loss

A continuación, se presentan los resultados obtenidos utilizando Margin Ranking Loss (MRL) como función de pérdida. Al igual que en el caso de Binary Cross Entropy, este modelo también utiliza una arquitectura GNN y un clasificador MLP. Sin embargo, en este caso, la diferencia clave radica en cómo se optimiza la red neuronal. Margin Ranking Loss se utiliza para entrenar el modelo en base a un margen de diferencia entre pares de nodos positivos y negativos.

En lugar de predecir simplemente la existencia de una relación entre dos nodos, como en BCE, Margin Ranking Loss se enfoca en ordenar correctamente los pares de nodos. La función de pérdida penaliza al modelo si a un par de nodos negativos se le asigna un score más alto que a un par de nodos positivos. Esto significa que el modelo no solo debe identificar correctamente si existe una relación, sino que también debe asegurar que las relaciones más fuertes (positivas) tengan un margen de puntuación suficientemente alto frente a las relaciones débiles (negativas).

Al igual que en el modelo anterior, el GNN genera embeddings para los nodos, pero en este caso, el objetivo de la función de pérdida es que el modelo aprenda a ordenar las relaciones de manera más precisa, asegurando que los nodos relacionados tengan una mayor puntuación que aquellos que no lo están. Este enfoque es especialmente útil para tareas donde la clasificación de los nodos y la precisión en el ranking es esencial.

Resultados desde Disorder → Drugs

Lo primero de todo será comprobar la relación DrugHasIndication, dada una enfermedad (disorder) como punto de partida se van a obtener los cinco fármacos con mayor puntuación de enlace para cada una. La tabla 4.6 muestra las 5 enfermedades para las que el modelo ha predicho los medicamentos con mayor score promedio en el top-5. En estos casos, el modelo estaría sugiriendo un posible tratamiento o reposicionamiento farmacológico, dependiendo de los datos que tenga de esas enfermedades. A continuación se van a analizar estos resultados

de forma clínica, ayudándonos de fuentes externas como con la configuración anterior.

Enfermedad	Score Promedio	Top-5 Fármacos
Alzheimer	0.8210	Fimasartan (0.8432) CLX-0921 (0.8264) S-P-Nitro ^a (0.8155) Indol-Bz ^b (0.8165) Practolol (0.8037)
Depressive disorder	0.8391	Corifollitropin alfa (0.8626) Fimasartan (0.8613) Trandolapril (0.8346) Nebivolol (0.8248) Cilazapril (0.8124)
Schizophrenia	0.8784	Fimasartán (0.9453) Cilazapril (0.8991) Trandolapril (0.8950) Losartán (0.8449) Isradipino (0.8078)
Nephrotic syndrome	0.8347	Acebutolol (0.8786) Risedronic acid (0.8342) Citrulline (0.8283) Papaverine (0.8270) Isradipine (0.8057)

Tabla 4.6: Top 5 medicamentos para el modelo GraphSAGE con Margin Ranking Loss

^a *S-P-Nitrobenzylloxycarbonylglutathione*

^b *(S)-N-(3-Indol-1-yl-2-methyl-propyl)-4-sulfamoyl-benzamide*

Para el Alzheimer, el modelo destaca cinco compuestos que encajan con los mecanismos clave de la enfermedad: Fimasartan, que puede mejorar el flujo sanguíneo cerebral, CLX-0921, un agente experimental con posible actividad antioxidante, S-P-Nitro, un derivado de glutatión que protege contra el estrés oxidativo, una indol-benzamida con potencial antiinflamatorio en el cerebro, y por último, recomienda Practolol, que es un betabloqueante que favorece la perfusión neuronal. En conjunto, estas cinco propuestas cubren circulación, defensa antioxidante e inflamación, tres ejes esenciales en el tratamiento contra el Alzheimer. A diferencia de la primera configuración que proponía sobre todo inhibidores clásicos de la ECA y moduladores de neurotransmisión, esta configuración aporta compuestos menos convencionales pero novedosos que servirían para explorar nuevas hipótesis terapéuticas.

Para el trastorno depresivo, el modelo sitúa en cabeza compuestos cardiovasculares: corifollitropina alfa, un modulador hormonal con posibles efectos indirectos sobre el estado de ánimo; fimasartán, un antagonista de los receptores de angiotensina; trandolapril y cilazapril,

inhibidores de la ECA que pueden influir en la inflamación y el estrés oxidativo; y nebivolol, un beta-bloqueante con perfil vascular y antioxidante. Estas predicciones sugieren que el modelo está realizando conexiones entre las alteraciones vasculares e inflamatorias con la depresión. Sin embargo, la implicación de un agente hormonal como la corifollitropina necesitaría validación de expertos. En contraste con la configuración anterior, centrada sobre todo en neuromoduladores y anticuerpos de acción directa en el cerebro, aquí se reflejan vías periféricas y vasculares que merece la pena explorar.

En el caso de la esquizofrenia, el modelo recomienda fármacos como fimasartán y losartán, dos antihipertensivos que bloquean la angiotensina y que pueden tener un efecto antiinflamatorio en el cerebro, algo cada vez más estudiado en trastornos psíquicos. A la vez, aparecen cilazapril y trandolapril, inhibidores de la enzima convertidora de angiotensina, que podrían ayudar a reducir el estrés oxidativo y la inflamación neuronal. Por último, isradipino, un bloqueador de canales de calcio, que sugiere un posible papel en la regulación de la liberación de neurotransmisores. Estas cinco propuestas están actualmente en el punto de mira para terapias complementarias en esquizofrenia, lo que refuerza la solidez de las predicciones del modelo.

Por último, para el síndrome nefrótico, el modelo sugiere compuestos muy coherentes para ayudar a retener proteínas y cuidar los pequeños vasos del riñón. Entre ellos están la isradipine (bloquea canales de calcio y baja la presión en los filtros renales), acebutolol (beta-bloqueante que regula el flujo de sangre en el riñón), papaverine (vasodilatador que mejora el riego), Citrulline (aumenta la producción de óxido nítrico para mantener saludables los vasos) y risedronate (bisfosfonato que podría reforzar la estructura del filtro glomerular). Aunque se usan en otras enfermedades, sus efectos encajan bien con las estrategias que buscan controlar la pérdida de proteínas y reducir la inflamación en el riñón.

Algunas de las conclusiones que podemos extraer con esta segunda configuración son las siguientes:

- **Énfasis en mecanismos periféricos y vasculares:** El modelo sugiere fármacos que actúan en órganos muy distintos y en vías que no sólo son del sistema nervioso, lo que refleja un entendimiento global de la biología de la enfermedad.
- **Ampliación del espacio de reposicionamiento:** Frente a la primera configuración, este modelo recomienda fármacos basados en compuestos cardiovasculares, inmunomoduladores o renales, ampliando el abanico de candidatos para reposicionamiento.

Resultados desde Drugs → Disorder

En este apartado se ha abordado el análisis desde la perspectiva inversa: partiendo de un fármaco, se ha evaluado qué enfermedades predice el modelo como posibles candidatas. Este enfoque es clave dentro del contexto del drug repurposing, ya que permite detectar nuevas

aplicaciones terapéuticas potenciales para compuestos ya existentes, lo cual puede ahorrar tiempo y costes en investigación clínica.

En la tabla siguiente 4.7 se muestran los medicamentos con sus respectivas enfermedades asociadas. Estas enfermedades alcanzan los valores de score más altos en su top-5, lo que refleja una alta confianza del modelo en esas asociaciones.

Fármaco	Score Promedio	Top-5 Enfermedades
Erythropoietin	0.8609	Retinal dystrophy (0.9115) Papillary renal cell carcinoma (0.8777) Anemia of prematurity (0.8610) Chronic liver failure (0.8377) Diabetic encephalopathy (0.8117)
Insulin glargine	0.8812	Hyperinsulinism due to UCP2 deficiency (0.9188) Schizophrenia 17 (0.9150) Mitochondrial myopathy with diabetes (0.8725) End stage renal failure (0.8563) Chronic liver failure (0.8438)
Palivizumab	0.8787	Thrombocythemia 1 (0.8910) Congenital adrenal hyperplasia (0.8901) Carey–Fineman–Ziter (0.8895) Keratoconus 1 (0.8777) Immunodeficiency 80 (0.8453)
Interferon alfacon-1	0.8462	Platelet abnormalities with eosinophilia (0.9042) Thrombocythemia 1 (0.8680) Juvenile idiopathic arthritis (0.8234) Immunodeficiency 84 (0.8204) Immunodeficiency 80 (0.8151)

Tabla 4.7: Top 5 enfermedades para el modelo GraphSAGE con Margin Ranking Loss

Lo más interesante de estos resultados es la posibilidad de descubrir nuevas indicaciones para fármacos ya conocidos, lo cual es uno de los pilares del drug repurposing. Algunos casos destacados, viendo la tabla 4.7 son los siguientes.

Para la eritropoyetina (EPO), una hormona producida principalmente por los riñones que estimula la formación de glóbulos rojos, el modelo coloca en los primeros puestos la retinopatía,

el carcinoma de células renales papilar, la anemia de la prematuridad, la insuficiencia hepática crónica y la encefalopatía diabética. Estas asociaciones tienen bastante sentido, ya que la capacidad de la EPO para estimular la formación de glóbulos rojos explica su papel en la anemia prematura y la insuficiencia hepática, mientras que sus efectos protectores sobre células neuronales y vasculares hacen que sea una candidata interesante para distrofias retinianas y complicaciones cerebrales de la diabetes.

El modelo muestra resultados interesantes para insulina glargina, destacando enfermedades donde el uso de insulina tiene sentido clínico evidente. Por ejemplo, identifica correctamente una alta relación con hiperinsulinismo por deficiencia de UCP2, una condición directamente vinculada a problemas de insulina. También sugiere asociaciones coherentes con trastornos metabólicos complejos, como la insuficiencia hepática crónica y la miopatía mitocondrial con diabetes, ambas con componentes importantes relacionados con resistencia a la insulina o diabetes. Por otro lado, la asociación con condiciones como la insuficiencia renal en etapa terminal refleja indirectamente problemas metabólicos comunes en pacientes renales. Más llamativa, sin relación clínica tan directa pero plausible desde la perspectiva metabólica, es la asociación con esquizofrenia 17, dado que la esquizofrenia está cada vez más relacionada con trastornos metabólicos y resistencia a la insulina. Estos resultados indican que el modelo es capaz de identificar conexiones útiles en contextos metabólicos amplios, aunque algunas predicciones específicas requieren validación adicional.

En cuanto a Palivizumab, un anticuerpo monoclonal utilizado tradicionalmente para prevenir infecciones respiratorias graves, el modelo identifica conexiones especialmente interesantes con trastornos inmunológicos y autoinflamatorios, que se alinean con su perfil terapéutico. La asociación con inmunodeficiencia 80 es especialmente llamativa y plausible, dado el contexto inmunológico. Además, condiciones como Trombocitemia 1 y Síndrome de Carey-Fineman-Ziter también sugieren posibles vías inflamatorias o inmunológicas comunes, lo cual podría abrir hipótesis relevantes para futuros estudios exploratorios de reposicionamiento terapéutico. Las predicciones relacionadas con condiciones estructurales como Queratocono y metabólicas como Hiperplasia suprarrenal congénita resultan menos evidentes y requerirían estudios adicionales para aclarar posibles vínculos biológicos o indirectos.

Los resultados obtenidos con Interferon alfacon-1 destacan por su coherencia biológica, destacando la utilidad del compuesto en trastornos relacionados con la inmunidad y procesos inflamatorios. Resaltan especialmente asociaciones plausibles como el trastorno plaquetario eosinofílico, un trastorno inmunitario con implicación inflamatoria significativa; así como trombocitemia 1, dado el conocido efecto del interferón en regular el exceso de plaquetas. También se observan relaciones razonables con patologías inmunológicas como la artritis idiopática juvenil y distintos tipos de inmunodeficiencias, lo que sugiere que el modelo está capturando adecuadamente el espectro terapéutico del interferón como inmunomodulador, fortaleciendo su valor como candidato a futuros estudios de reposicionamiento terapéutico.

En general, los resultados muestran que el modelo es capaz de recuperar enlaces razonables dentro del top-5 o top-10 para muchas entidades. Los resultados en su mayoría son promete-

dores y biológicamente plausibles, pero hay algún caso en el que el modelo realiza recomendaciones con scores altos que deben ser interpretados con precaución, ya que no tienen ningún respaldo médico y probablemente sean resultado de estructuras latentes del grafo biomédico más que de indicaciones terapéuticas directas.

4.4.3 GAT y Binary Cross Entropy

Por último, también se van a presentar los resultados obtenidos por las métricas aplicadas a este modelo. Este modelo se ha entrenado con arquitectura GNN usando GAT y clasificador MLP, empleando Binary Cross Entropy y los embeddings iniciales de BioBERT también han sido proyectados a una dimensión de 128.

Resultados desde Disorder → Drugs

A continuación analizaremos los resultados que se obtienen de la relación DrugHasIndication cuando se le da de entrada al modelo una enfermedad. Estos resultados se presentan en la tabla 4.8. Los resultados de este modelo demuestran una predicción más precisa en medicamentos relevantes para ciertas enfermedades, especialmente en términos de analgésicos y antiinflamatorios. Sin embargo, algunos de los fármacos predichos no son aplicables a las patologías específicas, lo que sugiere que el modelo sigue necesitando refinamientos.

En comparación con los modelos anteriores, este modelo muestra una tendencia a predecir fármacos más variados, lo que podría ser útil en enfermedades complejas o que dependan de más factores, pero la precisión en enfermedades altamente especializadas sigue siendo un área de mejora. En la tabla 4.8 vemos en más detalle estos resultados, para poder analizarlos uno a uno a continuación.

Con esta tercera configuración, los cinco fármacos que más destacan para Alzheimer son: Tributyrin, que estimula procesos de limpieza y protección en el cerebro. Fomepizol, que ayuda a reducir sustancias tóxicas que dañan las neuronas. Pyrazole, que podría calmar la inflamación en el cerebro. Arbutin, un compuesto con fuerte poder antioxidante para defender las células. Y por último el modelo también recomienda bisulfito de sodio, que contribuye a proteger la materia cerebral del desgaste. Si comparamos estos resultados con la primera configuración, donde se apostó por fármacos que mejoran el flujo sanguíneo en el cerebro, y con la segunda, que resaltó antidepresivos para estabilizar la química neuronal, aquí el modelo propone compuestos más orientados a limpiar y proteger las células cerebrales de los daños diarios.

En cuanto a la depresión, los cinco fármacos que recomienda el modelo son: Pimavanserin, un modulador de receptores de serotonina que puede equilibrar el ánimo. Dostarlimab, un anticuerpo que, aunque se usa en oncología, podría atenuar la inflamación cerebral asociada al estado de ánimo. Megestrol acetate, una progestina que influye en hormonas relacionadas con el apetito y la motivación. Dibenzeplin, un antidepresivo tricíclico clásico que restaura

Enfermedad	Score Promedio	Top-5 Fármacos
Alzheimer	0.7827	Tributyrin (0.7931) Fomepizole (0.7911) Pyrazole (0.7786) Arbutin (0.7785) Sodium bisulfite (0.7722)
Depressive disorder	0.7281	Pimavanserin (0.7508) Dostarlimab (0.7407) Megestrol acetate (0.7388) Dibenzepin (0.7019) Cycrimine (0.7087)
Schizophrenia	0.7116	Metildigoxina (0.7265) Esmolol (0.7159) Armodafinil (0.7098) Lorcaserin (0.7086) Sodium oxybate (0.6976)
Cirrhosis of liver	0.7168	Fomepizole (0.7295) Metyrapone (0.7165) Hyaluronic acid (0.7149) Agomelatine (0.7146) Oltipraz (0.7087)

Tabla 4.8: Top 5 medicamentos para el modelo GAT con Binary Cross Entropy

el equilibrio de neurotransmisores. Y por último cycrimine, un antiespasmódico con cierto efecto sedante que podría ayudar a reducir la ansiedad y mejorar el descanso. Si lo comparamos con la primera configuración, donde el modelo priorizó principalmente antidepresivos y fármacos que equilibran los niveles de neurotransmisores clave, como son Pimavanserin y Dibenzepin. Frente a la segunda configuración, que incluyó compuestos menos convencionales (como inmunomoduladores en trastornos pulmonares), aquí con este modelo, el énfasis vuelve a los mecanismos directamente relacionados con neurotransmisores y el control del estado de ánimo.

El modelo sugiere cinco fármacos poco convencionales para esquizofrenia, pero todos con lógica clínica. El primero que recomienda es Metildigoxina, que tiene efecto sobre el sistema nervioso autónomo. Esmolol, útil para controlar ansiedad y taquicardia. Armodafinil que sirve para mejorar la atención y la vigilia. Otro fármaco que recomienda es lorcaserin, que modula serotonina y podría aportar estabilidad del ánimo. Y por último, óxido de sodio que refuerza GABA⁴ y ayuda al sueño. A diferencia de la segunda configuración que apostaba por proteger la circulación cerebral, este modelo se centra en actuar sobre los síntomas cognitivos,

⁴GABA (ácido gamma-aminobutírico) es el principal neurotransmisor inhibitorio del sistema nervioso central; reduce la actividad neuronal y favorece la relajación y el sueño.

mejorando la concentración, el estado de ánimo y el patrón de sueño, abriendo nuevas vías de estudio.

Por último, en el caso de cirrosis hepática, el modelo propone propone fármacos con mecanismos viables clínicamente hablando: el Fomepizole, que reduce la acumulación de metabolitos dañinos del alcohol; el oltipraz, que muestra actividad antifibrótica en el hígado; la metyrapona que modula el cortisol y, por tanto, la inflamación. El modelo también recomienda el ácido hialurónico que favorece la regeneración de la matriz hepática, y la agomelatina que aporta efectos antioxidantes e inmunomoduladores.

Algunas de las conclusiones que podemos extraer con esta segunda configuración son las siguientes:

- **Enfoque en regulación neuroquímica y funciones cognitivas:** El modelo propone fármacos que inciden directamente sobre la atención, el ánimo y el sueño, lo que sugiere un aprendizaje preciso de las alteraciones cerebrales propias de cada trastorno.
- **Complementariedad con configuraciones previas:** Frente a las configuraciones anteriores, este modelo amplía el abanico hacia compuestos psicotrópicos y neuromoduladores, enriqueciendo así las hipótesis terapéuticas para reposicionamiento.

Resultados desde Drugs → Disorder

Los resultados obtenidos en esta relación inversa los podemos ver en la tabla 4.9. Estos resultados muestran un patrón en las predicciones que combina medicamentos que tienen efectos conocidos con otros medicamentos menos convencionales o fuera de las indicaciones estándar. Este enfoque refleja la capacidad del modelo para explorar nuevos medicamentos de una forma amplia, pero también se observa que, para que fuera un modelo más viable, se necesitaría hacer más ajustes para poder hacer predicciones más precisas clínicamente, en cuanto a la relación entre drugs y disorders. Analizando los resultados con más detalle, se puede argumentar lo siguiente.

Etanercept, un anticuerpo que bloquea el $\text{TNF}\alpha^5$, vuelve a destacar en patologías inflamatorias sistémicas: además de su fuerte vínculo con la depresión y los problemas de tensión arterial (hipotensión y miocardiopatía), mantiene altas puntuaciones en enfermedades hepáticas relacionadas con inflamación crónica y toxicidad. Aunque con el modelo de la primera configuración ya aparecieron indicios de usos antiinflamatorios y metabólicas, en esta tercera configuración el modelo refuerza de forma más clara las conexiones con el hígado y el sistema cardiovascular.

Para el caso de bivalirudin, al ser un anticoagulante potente, muestra una fuerte relación con efectos sobre el hígado (lesiones y acumulación de grasa) y con caídas de la presión

⁵ $\text{TNF}\alpha$ (factor de necrosis tumoral alfa) es una citocina proinflamatoria fundamental en la regulación de la respuesta inmune y la inflamación sistémica.

Fármaco	Score Promedio	Top-5 Enfermedades
Etanercept	0.9289	Depressive disorder (0.9324) Hypotensive disorder (0.9323) Cardiomyopathy (0.9306) MDSLD ^a (0.9253) Drug-induced liver injury (0.9242)
Bivalirudin	0.8884	Drug-induced liver injury (0.9014) Hypotensive disorder (0.8947) MDSLD ^a (0.8899) Depressive disorder (0.8892) Myocardial ischemia (0.8670)
Cetuximab	0.8998	Allergic rhinitis (0.9119) Depressive disorder (0.9044) Hypotensive disorder (0.9032) Cardiomyopathy (0.8927) Heart failure (0.8870)
Leuprolide	0.8718	Drug-induced liver injury (0.8840) Learning disability (0.8801) Hypotensive disorder (0.8681) Depressive disorder (0.8664) Sleep-wake disorder (0.8603)

Tabla 4.9: Top 5 enfermedades para el modelo GAT con Binary Cross Entropy

^a *Metabolic dysfunction-associated steatotic liver disease*

arterial. También aparece vinculado a isquemia cardíaca, lo que refleja su influencia en la circulación. Sorprende la alta puntuación en el trastorno depresivo, que podría relacionarse con podría deberse a alteraciones sistémicas secundarias a su acción vascular y al estrés que genera en el organismo.

En cuanto a cetuximab el modelo recomienda con altos scores a enfermedades como la rinitis alérgica, el trastorno depresivo, la hipotensión, la miocardiopatía y la insuficiencia cardíaca. La mayoría de estas asociaciones probablemente se deben a la proximidad topológica en el grafo (muchos nodos conectados a vías comunes) y no reflejan indicaciones terapéuticas reales. Solo la rinitis alérgica puede resultar plausible, dado que cetuximab bloquea el receptor del factor de crecimiento epidérmico⁶ (EGFR) y modula así la respuesta inflamatoria en la mucosa nasal. Este fármaco con la primera configuración, apuntaba a indicaciones oculares coherentes con su mecanismo frente a este modelo en el que las asociaciones son menos viables clínicamente hablando.

⁶El receptor del factor de crecimiento epidérmico (EGFR) es una proteína de membrana que regula la proliferación y la inflamación de las células epiteliales al unirse a sus ligandos.

Por último, vamos a analizar el fármaco leuprolide, un agonista de GnRH⁷ que aparece vinculado con lesiones hepáticas inducidas por fármacos, hipotensión y alteraciones del ciclo sueño-vigilia, todas ellas reacciones adversas conocidas de los agonistas de GnRH. La puntuación alta en depresión también encaja con cambios en el estado de ánimo descritos en pacientes tratados con Leuprolide, y aunque “learning disability” no es una indicación real, seguramente sea resultado de la cercanía en el grafo con nodos de desarrollo y función hormonal. Así, el modelo no solo captura posibles efectos secundarios clínicamente plausibles, sino que también muestra cómo las conexiones del grafo producen predicciones con fundamento biológico.

En resumen esta tercera configuración refuerza el potencial de reposicionamiento que ya vimos con las anteriores, destacando nuevas conexiones con enfermedades hepáticas, cardiovasculares e inflamatorias, así como con trastornos del ánimo. Aunque muchas de estas predicciones tienen sentido y apuntan a vías terapéuticas prometedoras, conviene recordar que algunas también surgen por la cercanía en el grafo de nodos que por una evidencia clínica sólida. Por ello, será clave contar con la revisión de expertos para validar y priorizar estas relaciones, aunque en general el modelo ofrece resultados muy alentadores para poder avanzar en futuros estudios de reposicionamiento. En la siguiente sección de conclusiones y líneas futuras se analizará de forma general estos resultados y los pasos que se podrían dar para seguir mejorando el proyecto.

⁷Los agonistas de la hormona liberadora de gonadotropina (GnRH) imitan la acción de la GnRH hipofisaria para reducir la producción de hormonas sexuales (estrógenos y testosterona)

Conclusiones y líneas futuras

Tras un análisis de los datos obtenidos a lo largo del trabajo, se concluye que el modelo desarrollado presenta resultados prometedores tanto desde un punto de vista cuantitativo como cualitativo en cuanto a las predicciones de enfermedades y de fármacos. No obstante, el modelo presenta puntos por mejorar y poder seguir trabajando en un futuro desarrollo y perfeccionamiento de la solución planteada para incrementar la precisión y viabilidad clínica del sistema propuesto.

En cuanto a los resultados obtenidos con los embeddings generados por BERT y BioBERT, se ha evidenciado que ambos modelos logran generar representaciones semánticas útiles y válidas. En particular, BioBERT ha obtenido resultados ligeramente superiores a BERT en términos de correlación con las relaciones biomédicas reales. El coeficiente de correlación de Spearman para los embeddings de BioBERT fue de 0.3929, frente al 0.3768 obtenido por BERT. Esta diferencia, aunque moderada, sugiere que BioBERT, al estar preentrenado con texto biomédico, es más eficaz para captar las relaciones específicas del dominio. No obstante, se destaca que BERT, aunque no esté especializado en el ámbito biomédico, también consigue ser útil para estas tareas, siempre que los textos estén bien estructurados y preparados adecuadamente.

En las tres configuraciones evaluadas: GraphSAGE con Binary Cross-Entropy (BCE), GraphSAGE con Margin Ranking Loss (MRL), y GAT con Binary Cross-Entropy (BCE), se ha observado un rendimiento competitivo en las métricas utilizadas para la predicción de enlaces. En términos generales, las métricas de AUC, Accuracy, Recall@k, Precisión@k, HitRate@k, MRR y NDCG@50 obtienen resultados altos. Destaca especialmente el modelo GraphSAGE con Margin Ranking Loss, que registra valores notables en AUC (0,86), Recall@k (0,94), MRR (0,93) y NDCG@50 (0,94), lo que confirma su capacidad para priorizar de manera efectiva relaciones relevantes entre nodos. En cuanto a la capacidad de priorizar enlaces relevantes, al utilizar un clasificador MLP, se ha conseguido obtener un rendimiento destacable en las tres configuraciones, sugiriendo que este tipo de clasificador tiene un impacto positivo en la precisión del modelo en tareas de ranking. En comparación con las configuraciones usadas con GraphSAGE, el modelo GAT, aunque es efectivo, ha mostrado un rendimiento ligeramente inferior en AUC (0,8235) y Precision@k (0,41), lo cual indica que los mecanismos de atención podrían no haber captado completamente la complejidad estructural del grafo en este contexto específico. Sin embargo, con GAT se han seguido obteniendo resultados muy competitivos en el resto de las métricas, lo que sugiere que los mecanismos de atención pueden ser útiles cuando se busca priorizar enlaces relevantes. Pese a estas pequeñas diferencias, todas las configuraciones han demostrado ser robustas, con valores de AUC que se mantienen por encima de 0.8 en todos los casos, lo que garantiza que el modelo tiene un buen nivel de discriminación entre pares de nodos relacionados y no relacionados.

Los resultados obtenidos de la predicción directa sobre el enlace Drug-Disorder muestran una tendencia interesante, ya que los modelos han logrado identificar asociaciones válidas entre

medicamentos y enfermedades, con puntuaciones (scores) generalmente altas, especialmente en el caso de la configuración de GraphSGAE con BCE. En la tercera configuración (GAT con BCE), las probabilidades asociadas a las predicciones específicas fueron un poco más conservadoras, situándose en torno a 0,75 y 0,8. Aún así todos los modelos obtienen resultados muy interesantes para futuros estudios de reposicionamiento farmacológico, detectando relaciones con fundamento clínico evidente, así como relaciones más indirectas posiblemente derivadas de efectos secundarios o proximidad estructural entre nodos en el grafo utilizado durante el entrenamiento.

Respecto a futuras líneas de investigación y mejora, se propone centrar esfuerzos en el refinamiento de los embeddings y en los datos iniciales. Una vía recomendable de actuación sería el fine-tuning de los embeddings. Aunque los embeddings de BioBERT han demostrado ser adecuados y tener unas métricas buenas, se podría intentar realizar un ajuste más fino sobre las representaciones, adaptándolas específicamente al dominio de las relaciones entre fármacos y enfermedades, ya que esto podría mejorar la capacidad predictiva en tareas específicas. Además, se podría experimentar con la incorporación de más datos, que cubran una mayor variedad de relaciones directas, enriqueciendo el contexto disponible para el modelo durante el entrenamiento, aunque ello supondría mayores exigencias computacionales.

Otra mejora potencial radica en el uso de técnicas de sampling más avanzadas durante el entrenamiento, como el “hard negative sampling”, para mejorar la capacidad del modelo al diferenciar relaciones complejas entre nodos positivos y negativos. También sería valioso considerar la posibilidad de integrar más información estructural en el grafo, lo que podría permitir que los modelos de GNN aprendan representaciones más detalladas y precisas de las interacciones entre nodos.

En términos generales, los resultados obtenidos validan que los modelos están adecuadamente diseñados y proporcionan predicciones útiles para la identificación de relaciones entre fármacos y enfermedades. Estas predicciones muestran un buen desempeño tanto en métricas cuantitativas como cualitativas, representando un buen punto de partida sólido para tareas como el reposicionamiento farmacológico. Sin embargo, es importante enfatizar que muchas de las asociaciones propuestas aún requieren validación clínica adicional, por lo que se recomienda utilizar estas predicciones como herramientas complementarias en fases exploratorias, siempre con supervisión médica experta y posterior validación experimental.

En conclusión, aunque existen aspectos susceptibles de mejora que podrían optimizar aún más el rendimiento y la precisión del modelo, los resultados obtenidos en este proyecto son prometedores y ofrecen una base sólida para investigaciones futuras. Las predicciones obtenidas actualmente son coherentes y se ajustan al contexto biomédico, demostrando que los modelos tienen la capacidad para identificar tanto relaciones evidentes como otras menos directas, lo que podría tener aplicaciones clínicas significativas en futuras tareas de reposicionamiento farmacológico.

Bibliografía

- [1] K. Latif and otros, “Drug reprofiling history and potential therapies against parkinson’s disease,” *Frontiers in Pharmacology*, vol. 13, 2022.
- [2] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio’, and Y. Bengio, “Graph attention networks,” *ArXiv*, vol. abs/1710.10903, 2017.
- [3] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” 2017.
- [4] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” 2018.
- [5] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, “Graph convolutional neural networks for web-scale recommender systems,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD ’18, ACM, July 2018.
- [6] P. Veličković and et al., “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2018.
- [7] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, “Temporal graph networks for deep learning on dynamic graphs,” 2020.
- [8] S. Yun, M. Jeong, R. Kim, J. Kang, and H. J. Kim, “Graph transformer networks,” in *Advances in Neural Information Processing Systems*, pp. 11983–11993, 2019.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018.
- [10] 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.
- [11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “Albert: A lite bert for self-supervised learning of language representations,” in *International Conference on Learning Representations*, 2020.

- [12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2019.
- [13] A. Name, “Graph neural networks series part 4: The gnns message passing & over-smoothing.” <https://medium.com/the-modern-scientist/graph-neural-networks-series-part-4-the-gnns-message-passing-over-smoothing-e77ffee523cc>, 2020. Accessed: 10-02-2025.
- [14] “Research and development.” <https://www.phrma.org/>. Accessed: 2024-03-26.
- [15] “What we do.” <https://www.fda.gov/about-fda/what-we-do>. Accessed: 2024-03-26.
- [16] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A comprehensive survey on graph neural networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4–24, 2019.
- [17] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, “How powerful are graph neural networks?,” *ArXiv*, vol. abs/1810.00826, 2018.
- [18] T. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *ArXiv*, vol. abs/1609.02907, 2016.
- [19] M. Ali, M. Berrendorf, C. T. Hoyt, L. Vermue, S. Sharifzadeh, V. Tresp, and J. Lehmann, “Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings,” *arXiv preprint arXiv:2007.14175*, 2020.
- [20] PyKEEN, “Pykeen: A python library for training and evaluating knowledge graph embeddings.” <https://github.com/pykeen/pykeen>, 2021.
- [21] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
- [22] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, “Xlnet: Generalized autoregressive pretraining for language understanding,” *arXiv preprint arXiv:1906.08237*, 2019.
- [23] jlmarketing54@gmail.com, “Gpt vs. bert: A comprehensive comparison of the leading language models.” <https://dailyaievolution.com/gpt-vs-bert-a-comprehensive-comparison-of-the-leading-language-models/>, 2023.
- [24] T. B. Brown *et al.*, “Language models are few-shot learners,” *arXiv preprint arXiv:2005.14165*, 2020.
- [25] K. S. Kalyan, “A survey of gpt-3 family large language models including chatgpt and gpt-4.” <https://arxiv.org/abs/2310.12321>, 2023. Preprint under review.

- [26] D. Chen, K. Lee, L. Wang, and W. Ammar, “Biomedlm: A domain-specific foundation model for biomedical natural language processing,” *arXiv preprint arXiv:2401.12049*, 2024.
- [27] S. Sadegh, J. Skelton, E. Anastasi, J. Bennett, D. B. Blumenthal, G. Galindez, M. Salgado-Albarrán, O. Lazareva, K. Flanagan, S. Cockell, *et al.*, “Network medicine for disease module identification and drug repurposing with the nedrex platform,” *Nature communications*, vol. 12, no. 1, pp. 1–12, 2021.
- [28] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, “Cytoscape: a software environment for integrated models of biomolecular interaction networks,” *Genome research*, vol. 13, no. 11, pp. 2498–2504, 2003.
- [29] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.
- [30] A. Name, “Inductive link prediction in knowledge graphs.” <https://towardsdatascience.com/inductive-link-prediction-in-knowledge-graphs-23f249c31961>, 2021. Accessed: 10-02-2025.
- [31] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [32] PyTorch, “torch.nn.marginrankingloss.” <https://pytorch.org/docs/stable/generated/torch.nn.MarginRankingLoss.html>, 2024. Accedido el 4 de mayo de 2025.
- [33] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.
- [34] C. Spearman, *The proof and measurement of association between two things*, vol. 15. 1904.
- [35] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [36] L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [37] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.
- [38] Y. Yang, R. N. Lichtenwalter, and N. V. Chawla, “Evaluating link prediction methods,” *Knowledge and Information Systems*, vol. 45, no. 3, pp. 751–782, 2015.
- [39] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

-
- [40] K. Järvelin and J. Kekäläinen, “Cumulated gain-based evaluation of ir techniques,” in *ACM Transactions on Information Systems (TOIS)*, vol. 20, pp. 422–446, ACM New York, NY, USA, 2002.
 - [41] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.
 - [42] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, “Introduction to recommender systems handbook,” in *Recommender Systems Handbook*, pp. 1–35, Springer, 2011.
 - [43] S. Pushpakom, F. Iorio, P. A. Eyers, K. J. Escott, S. Hopper, A. Wells, A. Doig, T. Guillelliams, J. Latimer, C. McNamee, *et al.*, “Drug repurposing: progress, challenges and recommendations,” *Nature Reviews Drug Discovery*, vol. 18, no. 1, pp. 41–58, 2019.
 - [44] European Group on Ethics in Science and New Technologies, “Statement on artificial intelligence, robotics and ’autonomous’ systems,” 2018.

Anexos

A Aspectos éticos, económicos, sociales y ambientales

A.1 Introducción

En la actualidad, la creciente complejidad de los sistemas sanitarios y la constante innovación tecnológica exigen métodos cada vez más avanzados para asegurar tratamientos eficaces y accesibles a todo el mundo. La aparición de nuevas enfermedades, sumada al envejecimiento de la población mundial y al incremento en los costes del desarrollo farmacéutico, ha impulsado la búsqueda de soluciones alternativas como el reposicionamiento de fármacos, estrategia que consiste en utilizar medicamentos ya aprobados para nuevas indicaciones terapéuticas. Esta práctica no solo acelera la disponibilidad de tratamientos efectivos para diversas patologías, sino que también reduce considerablemente los costes y tiempos asociados al desarrollo de nuevos medicamentos desde cero [43].

La inteligencia artificial, especialmente a través de técnicas avanzadas como las redes neuronales de grafos y el aprendizaje profundo, juega un papel fundamental en este escenario. Estos métodos permiten analizar grandes volúmenes de datos biomédicos para identificar relaciones novedosas entre medicamentos y enfermedades, facilitando así el descubrimiento de oportunidades terapéuticas que podrían pasar desapercibidas mediante métodos tradicionales. Dado la importancia que genera la inteligencia artificial en la era digital en la que vivimos, saber manejarse con ella es algo fundamental si se quiere destacar en el mercado.

Este trabajo se centra precisamente en la implementación de un sistema avanzado de recomendación médica basado en inteligencia artificial, específicamente redes neuronales de grafos, con el objetivo de identificar y proponer nuevas aplicaciones clínicas para fármacos existentes. Sin embargo, la introducción de tecnologías innovadoras en salud no está exenta de impactos potenciales, por lo que es fundamental analizar detalladamente sus implicaciones éticas, económicas, sociales y ambientales antes de su implementación generalizada.

A.2 Descripción de impactos relevantes relacionados con el proyecto

A.2.1 Impacto ético

El desarrollo de sistemas avanzados para la recomendación médica mediante redes neuronales de grafos (GNN) y modelos de lenguaje presenta importantes consideraciones éticas que deben ser abordadas de manera explícita desde el inicio del proyecto. Este trabajo se fundamenta en principios esenciales propuestos por organismos como el *European Group on Ethics in Science and New Technologies (EGE)* [44], entre los que destacan la transparencia en los métodos utilizados, la protección estricta de la privacidad y confidencialidad de los datos personales y médicos, así como garantizar la exactitud y fiabilidad de las recomendaciones generadas.

En particular, el uso de inteligencia artificial en salud, especialmente técnicas complejas como las redes neuronales de grafos, implica desafíos éticos específicos debido a su naturaleza inherentemente difícil de interpretar. Las GNN son modelos avanzados capaces de analizar relaciones complejas en grandes conjuntos de datos biomédicos, pero su estructura interna puede resultar opaca para usuarios y reguladores. Por ello, es esencial asegurar la transparencia y la explicabilidad de estos sistemas, permitiendo a los profesionales sanitarios y pacientes entender claramente cómo y por qué se llega a una recomendación concreta. Esta claridad es vital no solo para mantener la confianza pública y profesional en la herramienta, sino también para facilitar su adopción responsable en contextos clínicos reales.

Además, aunque este proyecto no utiliza directamente historiales clínicos detallados o información genética de pacientes, sí trabaja con bases de datos biomédicas complejas que incluyen información sobre fármacos, enfermedades, proteínas y genes. Aunque estos datos pueden no contener información personal directa, es igualmente crucial asegurar su tratamiento responsable y riguroso, cumpliendo con normativas éticas y legales aplicables como el Reglamento General de Protección de Datos (RGPD). Por ello, se deben implementar medidas que garanticen la integridad, seguridad y confidencialidad de estos datos en todas las etapas del proyecto, desde su obtención hasta su utilización en los modelos predictivos.

La justicia algorítmica constituye otro aspecto crucial en este contexto. Resulta fundamental evitar cualquier tipo de sesgo o discriminación que pueda perjudicar injustamente a ciertos grupos de pacientes debido a desequilibrios en los datos utilizados o sesgos inherentes a los algoritmos. En este sentido, se deben establecer procedimientos específicos para detectar, evaluar y mitigar posibles sesgos, asegurando recomendaciones médicas equitativas y justas.

Finalmente, aunque estos modelos de IA ofrecen grandes beneficios potenciales al ámbito sanitario, deben utilizarse siempre como herramientas complementarias al juicio clínico experto. Las decisiones clínicas no pueden delegarse completamente en sistemas automatizados; por el contrario, la supervisión continua por parte de profesionales sanitarios capacitados es indispensable. Esta supervisión crítica asegura no solo la calidad y seguridad clínica de las recomendaciones, sino también la alineación con estándares éticos elevados en la atención al paciente [44].

A.2.2 Impacto económico y social

El impacto económico de este proyecto es considerable, destacando especialmente por el potencial del reposicionamiento farmacológico asistido por inteligencia artificial (IA). Debido a que esta estrategia permite reducir significativamente los costes y tiempos asociados al desarrollo tradicional de nuevos medicamentos. Diversos estudios indican que la identificación de nuevas indicaciones terapéuticas para medicamentos ya aprobados podría recortar hasta en un 50 % los gastos iniciales de investigación y desarrollo [43]. Este ahorro económico no solo beneficia directamente a las compañías farmacéuticas, sino que también facilita el acceso a tratamientos eficaces de manera más rápida y asequible para los pacientes, al eliminar muchas de las fases más costosas y prolongadas del proceso convencional de desarrollo de fármacos.

Entre las ventajas económicas más destacables del reposicionamiento farmacológico mediante IA se encuentran:

- **Reducción considerable de costes:** Al reutilizar medicamentos ya aprobados, se evita el elevado coste económico asociado a las fases iniciales de descubrimiento y pruebas preclínicas.
- **Rapidez en la identificación de nuevas terapias:** Especialmente relevante en contextos críticos o emergentes, como pandemias, o para enfermedades raras con escasas opciones terapéuticas.
- **Ampliación del acceso terapéutico:** La reducción de costes facilita una mayor disponibilidad de tratamientos eficaces, beneficiando especialmente a sistemas sanitarios con recursos limitados.
- **Personalización de la atención médica:** La IA permite identificar tratamientos específicos que podrían adaptarse mejor a las necesidades individuales de cada paciente, mejorando así los resultados clínicos y reduciendo efectos adversos.
- **Apoyo a la decisión clínica:** Los modelos de IA proporcionan herramientas complementarias, no sustitutivas, que optimizan y aceleran la labor diaria de los profesionales sanitarios.

Además del impacto económico, los beneficios sociales derivados del proyecto también son significativos. La aplicación de estos sistemas tiene el potencial de ofrecer nuevas alternativas terapéuticas para enfermedades que actualmente cuentan con pocas o ninguna opción disponible, incluyendo patologías raras o emergentes. Al proporcionar tratamientos más precisos y personalizados, el sistema contribuye a mejorar la calidad asistencial, reduciendo riesgos asociados a la medicación y aumentando la eficacia general de los tratamientos.

Asimismo, es crucial resaltar que estos modelos no pretenden reemplazar a los profesionales sanitarios, sino complementar su labor, proporcionándoles herramientas que optimicen y apoyen sus decisiones clínicas cotidianas. Esto permite que los profesionales puedan emplear su

tiempo y atención en aspectos más críticos y sensibles de la atención sanitaria, tales como el diagnóstico diferencial, la evaluación de tratamientos complejos o la comunicación efectiva y empática con los pacientes.

Finalmente, el desarrollo y la implementación de estas soluciones tecnológicas pueden generar nuevas oportunidades laborales en el ámbito sanitario, especialmente relacionadas con la gestión, análisis avanzado e interpretación de datos biomédicos. De esta manera, se abre un campo laboral emergente, dinámico y con alto valor añadido que puede contribuir significativamente al crecimiento y mejora continua del sistema de salud.

A.2.3 Impacto medioambiental

Aunque un proyecto individual basado en inteligencia artificial como el que se ha desarrollado puede parecer inicialmente que tiene un impacto medioambiental limitado, es necesario analizarlo desde una perspectiva más amplia y considerando una posible implementación generalizada en contextos reales, como hospitales o centros médicos.

Durante el desarrollo de este proyecto, se utilizaron recursos computacionales relativamente básicos, principalmente dos ordenadores, lo cual implica una huella ecológica modesta a pequeña escala. Sin embargo, la puesta en práctica y el mantenimiento continuado de un sistema avanzado basado en redes neuronales de grafos, especialmente en entornos clínicos, requeriría infraestructuras tecnológicas más avanzadas como servidores específicos o GPUs especializadas para el entrenamiento de los modelos, cuya operación genera un consumo energético considerable y, por lo tanto, una huella de carbono relevante a largo plazo.

Por esta razón, si esta solución se implantase de manera extendida en hospitales y centros de investigación médica, sería crucial adoptar medidas sostenibles desde el inicio. Esto podría incluir optimizar el uso de recursos computacionales mediante código eficiente, utilizar modelos más ligeros que requieran menos poder de procesamiento, y favorecer infraestructuras tecnológicas sostenibles o alimentadas por energías renovables.

En definitiva, aunque la escala actual del proyecto tiene un impacto ambiental relativamente bajo, su potencial implementación generalizada exige considerar desde el principio prácticas sostenibles, garantizando así que el beneficio tecnológico y sanitario vaya siempre de la mano de un compromiso claro y efectivo con la protección medioambiental, adoptando prácticas que aseguren que el desarrollo tecnológico se realice siempre con responsabilidad ecológica.

A.3 Conclusiones

Considerando los aspectos éticos, económicos, sociales y medioambientales detallados en secciones anteriores, este proyecto presenta un impacto muy positivo si se implementa con responsabilidad.

Desde la perspectiva ética, resulta esencial mantener la transparencia, evitar sesgos algorítmicos, proteger adecuadamente los datos utilizados y asegurar siempre la supervisión clínica experta. Económica y socialmente, el reposicionamiento farmacológico asistido por inteligencia artificial puede reducir significativamente costes y tiempos en el desarrollo de nuevos tratamientos, facilitando así el acceso rápido y asequible a terapias eficaces para diversas enfermedades, incluyendo aquellas poco estudiadas o emergentes. En el ámbito medioambiental, aunque el impacto directo del proyecto individual es relativamente limitado, la adopción a gran escala de estas tecnologías en hospitales requeriría recursos energéticos importantes. Por ello, se recomienda implementar estrategias de sostenibilidad, optimizando los recursos computacionales y fomentando infraestructuras basadas en energías renovables.

En definitiva, gestionado correctamente y con una visión responsable, este proyecto ofrece amplias posibilidades para mejorar la atención médica y la calidad de vida de los pacientes, representando una valiosa herramienta de apoyo para los profesionales sanitarios y una contribución positiva para la sociedad.

B Presupuesto económico

COSTE DE MANO DE OBRA (coste directo)		Horas	Precio/hora	Total
		800	18 €	14.500,00 €
COSTE DE RECURSOS MATERIALES (coste directo)				
		Precio de compra	Uso en meses	Amortización (en años)
				Total
Ordenador personal (con Software)		800,00 €	12	5
Ordenador con GPU para entrenamiento de redes neuronales		4500,00 €	4	5
				160,00 €
				300,00 €
COSTE TOTAL DE RECURSOS MATERIALES				
				460,00 €
GASTOS GENERALES (costes indirectos)				
		15 %	sobre CD	2.244 €
BENEFICIO INDUSTRIAL		8 %	sobre CD+CI	1.376,32 €
SUBTOTAL PRESUPUESTO				
IVA APLICABLE			21 %	18.580,32 €
				3.901,87 €
TOTAL PRESUPUESTO				
				22.482,19 €