UNIVERSIDAD AUTÓNOMA DE CHIHUAHUA

FACULTAD DE INGENIERÍA

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CLASIFICACIÓN DE PELÍCULAS POR GÉNERO USANDO UNA ARQUITECTURA BASADA EN TRANSFORMERS

POR:

MARITRINI VELÁZQUEZ RUIZ

TESIS PRESENTADA COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRÍA EN INGENIERÍA EN COMPUTACIÓN

DIRECTOR DE TESIS: DR. JESÚS ROBERTO LÓPEZ SANTILLÁN

CHIHUAHUA, CHIH., MÉXICO. ABRIL 2026

Clasificación de Películas por Género Usando Una Arquitectura Basada en Transformers. Tesis presentada por Maritrini Velázquez Ruiz como requisito parcial para obtener el grado de Maestría en Ingeniería en Computación, ha sido aprobada y aceptada por:

M.I. Fabián Vinicio Hernández Martínez
Director de la Facultad de Ingeniería

M.I. Rodrigo de la Garza Aguilar
Secretario de Investigación y Posgrado

M.S.I. Karina Rocío Requena Yáñez
Coordinador Académico

Dr. Jesús Roberto López Santillán
Director de Tesis

Abril 2026

Comité:

Dr. Jesús Roberto López Santillán
Dr. Adrián Pastor López Monroy
Dra. Olanda Prieto Ordaz
Dra. Vania Carolina Álvarez Olivas

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

12 de mayo de 2026.

ING. MARITRINI VELAZQUEZ RUIZ
Presente. -

En atención a su solicitud relativa al trabajo de tesis para obtener el grado de Maestría en Ingeniería en Computación, nos es grato transcribirle el tema aprobado por esta Dirección, propuesto y dirigido por el director **Dr. Jesús Roberto López Santillán** para que lo desarrolle como Tesis, con el título **"Clasificación de películas por género usando una arquitectura basada en Transformers"**.

Índice de Contenido

UNIVERSIDAD AUTÓNOMA DE
CHIHUAHUA

5. Resultados

5.1. Comparación en MM-IMDB
5.2. Comparación en Moviescope
5.3. Efecto del aumento de datos
5.4. Discusión general de resultados

6. Conclusiones

6.1. Limitaciones y trabajo a futuro

Referencias

Currículum Vitae

A T E N T A M E N T E

"naturam subiecit aliis"

EL DIRECTOR

M.I. FABIÁN VINICIO HERNÁNDEZ MARTÍNEZ

FACULTAD DE INGENIERÍA U.A.CH.

DIRECCIÓN

SECRETARIO DE INVESTIGACIÓN Y POSGRADO

M.I. RODRIGO DE LA GARZA AGUILAR

Todo es mejor gracias a esa época horrible.
(Little Jesus, 2019)

Agradecimientos

Empiezo esta sección agradeciendo a mis padres, quienes han sido los mejores maestros que me pudo dar la vida. A Leonardo por apoyarme en este proceso desde el día uno.
Agradezco a mi alma máter, la Facultad de Ingeniería, en especial a todos los profesores con los que tomé clase.
A mis sinodales, por brindarme su apoyo en este trabajo de investigación.
Al doctor Roberto, por compartirme sus conocimientos, por su paciencia y por persuadirme a hacer el posgrado. Después vemos como minar cryptos.
A Almendrita, gracias por estos 16 años. Te amo montones.

Resumen

El aprendizaje automático multimodal se centra en métodos para modelar información procedente de más de una modalidad, como texto, imagen, audio y vídeo. Es fundamental que los modelos analicen y organicen simultáneamente diversas modalidades. Los recientes avances en las aplicaciones multimodales se han atribuido principalmente a la disponibilidad de nuevos conjuntos de datos multimodales a gran escala, al aumento de la capacidad computacional y a la mejora de las representaciones de las modalidades individuales.

Las plataformas digitales manejan enormes colecciones de películas, pero clasificarlas de forma automática sigue siendo un reto. Los modelos actuales suelen enfocarse solo en texto o imágenes y requieren muchos recursos.

Un modelo multimodal eficiente puede mejorar los sistemas de recomendación y la organización de catálogos de películas, ofreciendo experiencias más personalizadas a los usuarios. Además, al usar menos recursos, contribuye a un desarrollo tecnológico sostenible y accesible.

En este trabajo se propone una arquitectura multimodal basada en Transformers que combina un encoder visual ViT y un encoder textual BERT, ambos preentrenados, con un esquema de congelamiento parcial de capas y módulos de fusión mediante atención cruzada, además de un módulo de regularización para cada modalidad. Adicionalmente, se emplean técnicas de aumento de datos para ambas modalidades, utilizando back-translation para el texto y RandAugment para las imágenes, con el objetivo de mejorar la capacidad de generalización del modelo.

La arquitectura propuesta es evaluada en los conjuntos de datos MM-IMDB y Moviescope, utilizando métricas de F1 Score y micro/macro Average Precision para clasificación multi etiqueta.

Los resultados experimentales muestran que el modelo alcanza un desempeño competitivo frente a enfoques multimodales previamente reportados, superando a varios modelos de referencia y acercándose al estado del arte, a pesar de mantener un diseño más simple, destacando los resultados de 66.9 ± 0.1, 66.9 ± 02 y 67.1 ± 0.1 para $\mu - F1$, $W - F1$ y $s - F1$, respectivamente, en el conjunto de datos MM-IMDB; además de los resultados del 76.5 ± 0.6 en μAP, 74.8 ± 0.4 mAP, 78.3 ± 0.4 *weighted-AP* y 84.6 ± 0.4 *samples-AP*, para el conjunto de datos Moviescope.

Estos resultados sugieren que una adecuada combinación de encoders preentrenados, estrategias de fusión y aumento de datos puede ofrecer un balance efectivo entre rendimiento y complejidad del modelo.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Los recientes trabajos de aprendizaje profundo en tareas de clasificación de texto, análisis de sentimientos o reconocimiento de autores, como lo son los Transformers, han dado oportunidad a utilizar arquitecturas similares para el clasificado y generación de imágenes, siendo competitivas con el desempeño de las redes neuronales convolucionales que causaron un auge en procesamiento de imágenes. Ahora imaginemos que se puede enriquecer aún más las tareas de clasificación utilizando dos modalidades como imagen y texto, abriendo la puerta a los modelos multimodales, en los que se concatenan los vectores de imágenes con secuencias de palabras para darle más precisión a los resultados. Esto ya se puede apreciar en el apoyo de diagnósticos médicos [6], investigaciones de escenas de crimen [7] o el análisis de excavaciones de zonas arqueológicas, por mencionar algunos ejemplos.

Los modelos de lenguaje largos (LLMs, por sus siglas en inglés) y los modelos multimodales están siendo cada vez más utilizados para abordar estos problemas complejos. Los LLMs, como GPT-4 [8], son capaces de comprender y generar texto de manera coherente y contextualmente relevante, lo que los hace útiles para tareas que requieren una comprensión profunda del lenguaje natural. Estos modelos pueden generar descripciones detalladas, responder preguntas y realizar análisis textuales

avanzados, lo cual es crucial en aplicaciones como el subtitulado de imágenes.

En el caso de los modelos multimodales, su capacidad para procesar y entender diferentes tipos de datos simultáneamente los hace ideales para tareas que combinan texto e imágenes. Modelos como DALL-E [9], que pueden generar imágenes a partir de descripciones textuales, demuestran el potencial de estas tecnologías para la creación de contenido visual a partir de información textual. Estos modelos no solo entienden el contenido visual, sino que también pueden relacionarlo con descripciones textuales, permitiendo una integración fluida entre diferentes modos de información.

Si bien los modelos basados en Transformers han demostrado un alto desempeño en tareas de generación y clasificación tanto de texto como de imágenes, su entrenamiento completo suele implicar un costo computacional elevado. Esto ha motivado la exploración de estrategias que permitan aprovechar el conocimiento de modelos preentrenados sin necesidad de ajustar todos sus parámetros. Una alternativa común consiste en congelar parcialmente los encoders preentrenados y entrenar únicamente los módulos añadidos para una tarea específica, reduciendo así el tiempo de entrenamiento y los recursos requeridos.

En este trabajo se adopta este enfoque, utilizando BERT (Bidirectional Encoder Representations from Transformers) y ViT (Vision Transformer) como encoders preentrenados, manteniendo fijas las primeras capas de cada modelo y ajustando únicamente los módulos de fusión y clasificación. A diferencia del ajuste fino completo, donde todos los parámetros del modelo se actualizan, esta estrategia permite conservar las representaciones generales aprendidas durante el preentrenamiento y centrar el aprendizaje en la integración de las modalidades textual y visual, logrando un balance entre eficiencia computacional y desempeño.

Adicionalmente, con el objetivo de mejorar la capacidad de generalización del modelo y moderar los efectos del desbalance entre clases presente en los conjuntos

de datos utilizados, se incorporaron técnicas de aumentación de datos tanto en la modalidad textual como visual. Para el texto, se empleó una estrategia de *back-translation*, la cual permite generar variantes semánticamente consistentes de las descripciones originales sin alterar su significado. En el caso de las imágenes, se aplicaron transformaciones aleatorias controladas sobre los pósters de películas, con el fin de incrementar la diversidad visual durante el entrenamiento. Estas técnicas permiten enriquecer los datos disponibles sin incrementar el costo de adquisición, contribuyendo a un aprendizaje más robusto del modelo.

El objetivo de esta investigación trata de diseñar un modelo novedoso que utilice arquitecturas basada en Transformers , tanto en codificación, como atención cruzada y congelamiento de capas, para la clasificación de imágenes con su respectiva descripción textual, utilizando una base de datos que contiene pósters de películas y la sinopsis de esta misma. Esta arquitectura busca ser competitiva con el estado del arte, al tiempo que propone un diseño modular y menos complejo que otras arquitecturas multimodales previamente reportadas.

1.1. Justificación

Las películas siguen siendo una forma de entretenimiento muy popular y la clasificación de estas mismas por su género se ha convertido en una tarea de gran interés ya que, con el incremento de servicios de streaming, como Netflix, Prime Video, entre otros, hace que se crea y distribuya en línea una gran cantidad de datos sobre películas. A menudo, las plataformas de streaming optan por clasificar su catálogo de películas por algoritmos que se alimentan de los metadatos de las mismas, o por recursos humanos. Cada película debe ser cuidadosamente revisada y analizada, lo cual consume una cantidad considerable de tiempo y esfuerzo por parte del equipo

encargado. Este proceso manual es propenso a errores humanos, como omisiones, errores tipográficos o inconsistencias en la información proporcionada.

La importancia de una adecuada clasificación de películas por género radica en que permite mejorar los sistemas de recomendación utilizados en plataformas digitales. Una categorización más precisa facilita la generación de sugerencias acordes a los intereses y preferencias de los usuarios, lo que contribuye a una experiencia de navegación más relevante y satisfactoria.

Desde la perspectiva del usuario, una mejor clasificación permite acceder de forma más rápida a contenido afín a sus gustos, evitando recomendaciones poco relevantes y facilitando la exploración de nuevos títulos dentro de sus géneros de interés. Esto resulta especialmente importante en contextos donde existe una gran cantidad de contenido disponible.

Por otro lado, para las plataformas, una organización más eficiente del catálogo contribuye a optimizar la presentación del contenido y a mejorar la calidad de sus sistemas de recomendación. En conjunto, esto favorece tanto la experiencia del usuario como la gestión del contenido, promoviendo un acceso más adecuado y personalizado a la información.

Utilizando métodos de automatización con aprendizaje profundo y aprovechando la multimodalidad del conjunto de datos propuesto que utiliza el póster y la sinopsis de la película, como imagen y texto respectivamente, se analizan las características individuales de cada modalidad, para crear un clasificador rico en información y acertado en sus resultados. Hasta la fecha no se ha encontrado una publicación científica que use un modelo con arquitecturas de ViT Transformer y BERT usando atención cruzada, congelamiento de capas de los codificadores y drop out por modalidades, lo que da espacio a un modelo novedoso en el área de clasificación de contenidos de streaming.

1.2. Objetivos

1.2.1. Objetivo General:

Diseñar un Modelo de Red Neuronal basado en la arquitectura de Transformers, usando técnicas de congelamiento de capas, normalización y regularización por modalidades para su entrenamiento, que sea capaz de obtener resultados competitivos con el estado del arte, para un problema de clasificación multimodal, específicamente utilizando imágenes y texto.

1.2.2. Objetivos Específicos:

1. Analizar los diferentes modelos computacionales que se han utilizado en trabajos anteriores para abordar este tipo de problemáticas.

2. Obtener y acondicionar los conjuntos de datos multimodales para entrenar los modelos que se elegirán.

3. Diseñar el nuevo modelo y entrenarlo con técnicas de congelamiento de capas y un encoder de Transformer para su clasificación, con el fin de aprender representaciones conjuntas de texto e imagen para la clasificación multi etiqueta de géneros de cine.

4. Comparar el modelo propuesto con los del estado del arte, mediante métricas estándar de clasificación multi etiqueta como F1-score y Average Precision, en los conjuntos de datos MM-IMDB y Moviescope.

5. Analizar el impacto de las técnicas de aumento de datos en ambas modalidades, utilizando back-translation para texto y RandAugment para imágenes, en el desempeño del modelo propuesto.

1.3. Hipótesis

H1: Un modelo de aprendizaje profundo basado en encoder de Transformers y congelamiento de capas en colecciones de datos multimodales, sobre películas y su descripción gráfica y textual (póster y sinopsis), obtiene resultados competitivos con el estado del arte en tareas de clasificación multi etiqueta, utilizando una arquitectura sencilla y modular.

Capítulo 2

Trabajos Relacionados

El artículo titulado “Learning Transferable Visual Models from Natural Language Supervision” [10] aborda un tema central en la intersección de la visión por computadora y el procesamiento del lenguaje natural (NLP). La investigación explora la posibilidad de trasladar los avances logrados en el NLP, mediante el pre entrenamiento a gran escala con supervisión textual, al campo de la visión por computadora. El problema clave que plantea la investigación es, si los métodos que aprenden directamente de grandes volúmenes de texto en bruto podrían desencadenar un avance similar en la visión por computadora, tradicionalmente dominada por modelos pre entrenados en conjuntos de datos etiquetados manualmente, como ImageNet.

El objetivo de la investigación es demostrar que la tarea de predecir qué descripción textual se asocia a una imagen, es una forma eficiente de aprender representaciones visuales transferibles. Para ello, los investigadores crearon Çontrastive Language-Image Pre-training ” (CLIP), un modelo que combina un codificador de texto y un codificador de imágenes entrenado para predecir el vínculo entre imágenes y descripciones en un conjunto de datos masivo de 400 millones de pares de imágenes y textos extraídos de internet.

Este enfoque plantea una hipótesis simple pero poderosa: el aprendizaje supervisado por lenguaje natural puede ser más eficaz que los métodos tradicionales basados en etiquetas manuales. La metodología empleada se basa en el uso de codificadores duales, donde tanto imágenes como textos son procesados por modelos separados pero entrenados conjuntamente. El modelo no está limitado a un dominio visual particular, ya que puede sintetizar un clasificador zero-shot, lo que significa que puede realizar tareas de clasificación sin necesidad de ver ejemplos etiquetados del conjunto de datos específico. Los experimentos se realizaron en varios conjuntos de datos como STL10, CIFAR10 , y Food101, mostraron que CLIP no solo puede competir en tareas de clasificación tradicionales, sino que también ofrece capacidades de transferencia zero-shot.
Los resultados principales indican que CLIP supera el 90 % de precisión en varios conjuntos de datos, tanto en configuraciones de zero-shot como de supervisión completa, lo que refuerza la hipótesis de que el modelo puede transferir eficazmente su conocimiento a nuevas tareas. Esto es particularmente revelador, ya que tradicionalmente los modelos de visión por computadora han dependido de conjuntos de datos manualmente etiquetados y de técnicas de ajuste fino específicas para cada tarea.

Utilizando meta-aprendizaje basándose en la arquitectura de CLIP, tenemos el trabajo de Ferragu et. al [11] , en el cual se demuestra que la combinación de las modalidades de los codificadores de texto e imagen de CLIP supera a los aprendices meta-few-shot más avanzados en parámetros de referencia ampliamente adoptados, todo ello sin formación adicional. Nuestros resultados confirman el potencial y la solidez de los modelos de fundamentos multimodales como CLIP y sirven de referencia para los enfoques existentes y futuros que aprovechan este tipo de modelos. Probándose con los conjuntos de datos CIFAR-FS y miniImageNet, se tuvo

una mejoría en relación con el estado del arte anterior obteniendo un 98 % y 99 % de accuracy con 1 y 5 ejemplos, usando solo 5 clases.

Por su parte el artículo “Your Diffusion Model is Secretly a Zero-Shot Classifier” por Li et al [12] exploran cómo los modelos de difusión de texto a imagen, como Stable Diffusion, pueden utilizarse para tareas de clasificación sin necesidad de entrenamiento adicional. Tradicionalmente, estos modelos se han empleado principalmente para la generación de imágenes a partir de texto. Sin embargo, los autores demuestran que las estimaciones de densidad condicional generadas por estos modelos pueden aprovecharse para realizar clasificación en escenarios de "zero-shot", es decir, sin entrenamiento previo específico para la tarea.

La metodología propuesta, denominada "Diffusion Classifier", utiliza las estimaciones de densidad de los modelos de difusión para asignar etiquetas a imágenes de manera efectiva. Este enfoque generativo para la clasificación ha mostrado resultados sólidos en diversos conjuntos de datos y supera a métodos alternativos que intentan extraer conocimiento de los modelos de difusión. Además, se destaca que, aunque persiste una diferencia de rendimiento entre los enfoques generativos y discriminativos en tareas de reconocimiento "zero-shot", el método basado en difusión exhibe una capacidad de razonamiento composicional multimodal más robusta que los enfoques discriminativos tradicionales.

Adicionalmente, los autores aplican el "Diffusion Classifier"para extraer clasificadores estándar de modelos de difusión entrenados de manera condicional en clases específicas, como aquellos entrenados en ImageNet. Estos modelos alcanzan un rendimiento de clasificación notable utilizando únicamente aumentaciones débiles y muestran una robustez cualitativamente superior ante cambios en la distribución de los datos. En conjunto, los hallazgos de este estudio representan un avance hacia la utilización de modelos generativos en tareas tradicionales de clasificación, sugiriendo que los modelos de difusión pueden servir como clasificadores efectivos sin

requerir entrenamiento adicional. El modelo alcanzó una precisión del 79.1 % en ImageNet utilizando solo aumentos débiles y muestra una mayor solidez al cambio de distribución que los clasificadores discriminativos de la competencia entrenados en el mismo conjunto de datos. Los resultados sugieren que puede haber llegado el momento de revisar los enfoques generativos de la clasificación.

En el trabajo de Arevalo et al, del 2017, "GATED MULTIMODAL UNITS FOR INFORMATION FUSION" [13], introducen las GMU, una estrategia basada en redes neuronales para la clasificación multi etiqueta de datos multimodales. Inspirado en el control de flujo en arquitecturas recurrentes como GRU (Gated Reccurent Units) o LSTM (Long Short-Term Memory). Una GMU está pensada para ser utilizada como unidad interna en una arquitectura de red neuronal cuyo propósito es encontrar una representación intermedia basada en una combinación de datos de diferentes modalidades. El modelo de compuerta puede reutilizarse en diferentes arquitecturas de red para resolver distintas tareas, y puede optimizarse de extremo a extremo con otros módulos de la arquitectura, utilizando algoritmos estándar de optimización basados en gradientes.
Como caso de aplicación, exploran la tarea de identificar el género de una película basándose en su argumento y su cartel, utilizando el dataset que ellos mismos crearon, MultiModal IMBDB (MM-IMDB), el cual consisten en más de 25 mil películas con su respectivo póster y sinópsis.
Para la representación textual, se usó la técnica Word2Vec para la creación de embeddings (espacios vectoriales representativos). Por su parte, en la representación visual, se utilizaron dos enfoques: el primero consistió en utilizar VGG como extractor de características. Este modelo se denomina transferencia VGG. El segundo enfoque toma como entrada las imágenes en bruto para una CNN. El clasificador del modelo se basó en un perceptrón multicapa (MLP) con dos capas totalmente

conectadas y función de activación maxout. Los resultados de estos experimentos fueron con las métricas de F1, teniendo así 61.7 para el F1-weighted, 63.0 en F1-samples y F1-micro, y por último para F1-macro 54.1.

El artículo "Supervised Multimodal Bitransformers for Classifying Images and Text"de Kiela et al. [14] presenta un modelo llamado Multimodal Bitransformer , MMBT, que integra información de texto e imágenes para tareas de clasificación. Este enfoque combina codificadores pre entrenados de texto e imagen, fusionando sus representaciones en el espacio de tokens de BERT, lo que permite una adaptación más sencilla a avances unimodales sin requerir reentrenamiento multimodal. El MMBT ha demostrado un rendimiento competitivo en diversas tareas de clasificación multimodal, incluyendo conjuntos de datos como MM-IMDB, Food101 y V-SNLI, superando a varios métodos de fusión alternativos y, en algunos casos, equiparando a modelos multimodales pre entrenados más complejos como ViLBERT.
Se encontró que el Bitransformer multimodal supera a las bases por un margen significativo. En la fusión tardía, FiLMBert y ConcatBert obtienen resultados similares. Especulamos que la causa de la mejora de MMBT sobre ConcatBert es su capacidad para permitir que la información de diferentes modalidades interactúe a diferentes niveles, a través de la autoatención, en lugar de sólo en la capa final. Parte de la mejora procede del rendimiento superior de BERT (lo que tiene sentido, dado el predominio del texto), pero incluso así MMBT mejora a BERT en, por ejemplo, $\sim$ 3 % en MM-IMDB Macro-F1. 61.6$\pm$.2/66.8$\pm$.1

En el trabajo de Moreno-Galván et al. titulado “Automatic movie genre classification & emotion recognition via a BiProjection Multimodal Transformer”[15] introduce un modelo llamado BiProjection Multimodal Transformer diseñado para

clasificar automáticamente los géneros de películas y reconocer las emociones que evocan. Este enfoque multimodal integra información de diversas fuentes, como audio, video y texto, para capturar de manera efectiva las características distintivas de cada película.

El modelo emplea una arquitectura de Transformer con biproyección que permite la interacción y fusión eficiente de las diferentes modalidades de datos. Esta estructura facilita la extracción de patrones complejos y la representación conjunta de las características multimodales, mejorando la precisión en la clasificación de géneros y el reconocimiento de emociones.

Los experimentos realizados demuestran que el BiProjection Multimodal Transformer supera a métodos anteriores en tareas de clasificación de géneros cinematográficos y reconocimiento de emociones, destacando su capacidad para manejar la complejidad y diversidad de los datos multimodales presentes en las películas. Este trabajo aporta una contribución significativa al campo de la fusión de información y al análisis automático de contenido multimedia, ofreciendo una herramienta eficaz para la categorización y comprensión de materiales audiovisuales. En los resultados que se tuvieron con el conjunto de datos MM-IMDB, superó al estado del arte en un 1 %, obteniendo un F1-micro de 68.9, F1-macro de 58.7, F1-weighted de 68.8 y F1-samples de 69.4.

La Investigación “Expanding Large Pre-trained Unimodal Models with Multimodal Information Injection for Image-Text Multimodal Classification” de Liang et al. [16], aborda la expansión de modelos preentrenados unimodales, como DenseNet y BERT, para tareas de clasificación multimodal de imágenes y texto. Los autores proponen el uso de un complemento denominado "Multimodal Information Injection Plug-in” (MI2P), que se integra en diferentes capas de los modelos unimodales. Este complemento permite la transformación de características entre modalidades al aprender correlaciones detalladas entre las características visuales y textuales.

De esta manera, se inyecta información textual en la red visual y viceversa, sin necesidad de modificar la estructura original de los modelos. Esta metodología busca aprovechar la capacidad de procesamiento intra-modal de los modelos pre entrenados, al tiempo que facilita una interacción efectiva entre modalidades para mejorar el rendimiento en tareas de clasificación multimodal. Para la parte visual, utilizan DenseNet como extractor de características de imágenes. En la parte textual, emplean BERT como modelo de lenguaje. Se introduce el módulo MI2P en diferentes capas de los modelos pre entrenados. Este módulo transforma características de una modalidad en un espacio latente que permite la transferencia de información entre texto e imagen.

La inyección se realiza en un esquema bidireccional:

- Imagen $\rightarrow$ Texto: Se agregan características visuales relevantes al modelo de texto.
- Texto $\rightarrow$ Imagen: Se integran características textuales al modelo visual.

La integración de MI2P permite que los modelos pre entrenados sigan aprovechando sus capacidades unimodales, mientras que el módulo facilita el aprendizaje de correlaciones multimodales. Se usa un enfoque de aprendizaje supervisado con funciones de pérdida específicas para optimizar la combinación de características. Sus resultados, en base a las métricas de F1-macro y F1-micro fueron del 64.2 y 70.8, respectivamente.

En el trabajo de investigación de Pahde F. et. al, de 2021 llamado "Multimodal Prototypical Networks for Few-shot Learning" [17] se presentan un marco de generación de características intermodales capaz de enriquecer el espacio de embeddings poco poblado en escenarios de pocos datos, aprovechando los datos de la modalidad auxiliar. Se entrenó un modelo generativo que mapea los datos de texto en el espacio de características visuales para obtener prototipos más fiables. Esto permite

explotar datos de modalidades adicionales (por ejemplo, texto) durante el entrenamiento, mientras que la tarea final en el momento de la prueba sigue siendo la clasificación con datos exclusivamente visuales.
Se utiliza una representación de K vecinos cercanos para la parte textual del encoder, y una arquitectura basada en Redes Residuales, ResNet-18 para la parte visual, luego se crea una Red GAN con una parte generadora y otra discriminatoria. El marco GAN, que contiene un generador Gt y un discriminador D, está optimizado para transformar una incrustación de texto dada por un codificador de texto preentrenado en el espacio de incrustación visual ϕ producido por un codificador de imagen preentrenado. El discriminador calcula una pérdida de reconstrucción (real/falsa) y una pérdida de clasificación auxiliar.
Los experimentos se probaron con el Dataset CUB-200 y Oxford-102, compitiendo con modelos unimodales de few shot learning más avanzados de ese momento (MAML, Matching Networks, Meta-Learn LSTM, etc), logrando tener una superioridad de accuracy con 75 % para 1 ejemplo y 83.50 % en 5 ejemplos.

En el artículo "Dynamic Regularization in UDA for Transformers in Multimodal Classification"[18] de 2023, presenta dos contribuciones principales para superar desafíos en el aprendizaje multimodal, específicamente para la clasificación de géneros de películas usando texto e imagen.
Primero, introducen un modelo de dos flujos llamado Multimodal BERT-ViT (MMBV), que conecta dos transformadores (BERT para texto y ViT para imagen) mediante la fusión dinámica de sus tokens clasificadores CLS (Classify Token), logrando una fusión efectiva y eficiente sin necesidad de módulos complejos adicionales. Esto mejora la integración de la información del texto (la modalidad fuerte) con la imagen (la modalidad más débil). Adaptan el marco de entrenamiento semi-supervisado Unsupervised Data Augmentation (UDA) con una estrategia de regularización di-

námica para equilibrar la especialización y generalización del modelo y evitar el sobreajuste. Esta regularización ajusta dinámicamente la influencia de la pérdida de consistencia durante el entrenamiento para que el modelo no se especialice excesivamente ni subentrene.
Evaluaron sus propuestas en las tareas de clasificación multietiqueta en los conjuntos de datos Moviescope y MM-IMDb, mostrando mejoras significativas en el desempeño de ambos modelos, especialmente en la capacidad para aprovechar mejor la modalidad débil sin perder información de la modalidad fuerte. La regularización dinámica demostró ser más efectiva que la UDA con parámetros fijos para controlar el sobreajuste. Registarn resultados de micro F1 de 66 % en el dataset de IMDB y de 74 % de micro Average Precision para el Moviescope.

Por su parte, en el artículo publicado en 2022, "LoRA: Low-rank adaptation of large language models" de Hu et al. [19], proponen una técnica que congela los pesos del modelo preentrenado e inyecta matrices de descomposición de bajo rango entrenables en cada capa de la arquitectura Transformer. Esto reduce drásticamente el número de parámetros entrenables para tareas posteriores. En GPT-3, LoRA puede reducir los parámetros entrenables en 10,000 veces y disminuir los requisitos de hardware de cómputo en 3 veces en comparación con el ajuste fino completo.
A pesar de tener menos parámetros entrenables, una mayor eficiencia en el entrenamiento y sin latencia adicional en la inferencia, LoRA logra un rendimiento comparable o superior al ajuste fino tradicional tanto en GPT-3 como en GPT-2. Es un método de ajuste fino mejorado en el que, en lugar de ajustar todos los pesos que constituyen la matriz de pesos del modelo de lenguaje grande entrenado previamente, se ajustan dos matrices más pequeñas que se aproximan a esta matriz más grande. Estas matrices constituyen el adaptador LoRA. Este adaptador ajustado se carga luego en el modelo entrenado previamente y se utiliza para la inferencia.

El resultado en estas pruebas con GPT-3 utilizando 37.7 millones de parámetros entrenables, arroja una exactitud del 73.8 % para el conjunto de datos WikiSQL y un 91.7 % en MNLI-m, comparado con el GPT-3 original (usando 175 billones de parámetros) en el que se obtuvo una exactitud de 73 % y 89.5 % respectivamente.

Capítulo 3

Marco Teórico

Esta sección se dedicará a dar un preambulo a los conceptos necesarios para el diseño de los algoritmos propuestos.

3.1. Inteligencia Artificial

La inteligencia artificial nació en la década de 1950, cuando un puñado de pioneros del naciente campo de la informática empezaron a preguntarse si se podía hacer que las computadoras "pensaran", una cuestión cuyas ramificaciones se sigue explorando hoy en día. De forma concisa, la Inteligencia Artificial puede describirse como el esfuerzo por automatizar tareas intelectuales normalmente realizadas por humanos [20].

3.2. Aprendizaje Automático

Un subconjunto de la inteligencia artificial (IA), el aprendizaje automático, o Machine Learning, es el área de la ciencia computacional que se centra en el análisis y la interpretación de patrones y estructuras de datos que hacen posible el aprendizaje, el razonamiento y la toma de decisiones sin interacción humana.

El aprendizaje automático se trata de extraer conocimiento de los datos. Es un campo de investigación en la intersección de la estadística, la inteligencia artificial y la informática y también es conocido como análisis predictivo o aprendizaje estadístico [21].

Dicho de otro modo, el aprendizaje automático trata de resolver problemas basándose en ejemplos históricos, en el que se implica el aprendizaje en patrones ocultos y su posterior utilización para clasificar o predecir un evento relacionado con el problema.

3.2.1. Aprendizaje Supervisado y no Supervisado

El aprendizaje automático supervisado implica un atributo de salida predeterminado, además del uso de atributos de entrada. Los algoritmos intentan predecir y clasificar el atributo predeterminado, y su precisión y clasificación errónea, junto con otras medidas de rendimiento, depende de los recuentos del atributo predeterminado o clasificado correctamente. El aprendizaje supervisado simplemente formaliza la idea de aprender por medio de ejemplos. [21]

En esta rama, el algoritmo que aprende obtiene dos conjuntos de datos, uno de entrenamiento y otro de prueba. Cuando se está entrenando un algoritmo de aprendizaje supervisado, los datos de entrenamiento van a consistir en entradas emparejadas con las salidas correctas, durante este tiempo, se buscarán patrones que se correlacionen con los resultados deseados; al pasar a las pruebas, el programa recibirá entradas no vistas y determinará la etiqueta con la que se clasificarán las nuevas entradas, basándose en los resultados obtenidos en el entrenamiento.

Entre los algoritmos de aprendizaje supervisado más comunes para las tareas de clasificación se encuentran los clasificadores lineales, máquinas de vectores de soporte, árboles de decisión, bosque aleatorio y k-vecinos cercanos.

Por el contrario, el aprendizaje de datos no supervisado implica el reconocimiento

de patrones sin la de un atributo objetivo. Todas las variables utilizadas en el análisis se utilizan como entradas y, debido al enfoque, las técnicas son adecuadas para la agrupación y técnicas de minería de asociaciones. Los algoritmos de aprendizaje no supervisado son adecuados para crear las etiquetas en los datos que posteriormente se utilizan para ejecutar tareas de aprendizaje supervisado. [22]

En algunos problemas de reconocimiento de patrones, los datos de entrenamiento consisten en un conjunto de vectores de entrada “X"sin ningún valor objetivo correspondiente. El objetivo en estos problemas de aprendizaje no supervisado puede ser descubrir grupos de ejemplos similares dentro de los datos, lo que se denomina agrupación, o determinar cómo se distribuyen los datos en el espacio, lo que se conoce como estimación de la densidad.

3.3. Aprendizaje Profundo

El aprendizaje profundo permite que los modelos computacionales que se componen de múltiples capas de procesamiento, aprendan representaciones de datos con múltiples niveles de abstracción. Estos métodos han mejorado drásticamente el estado del arte en el reconocimiento de voz, el reconocimiento de objetos visuales, la detección de objetos y muchos otros dominios, como el descubrimiento de fármacos y la genómica.

El término "profundo” en ”aprendizaje profundo” no se refiere a ningún tipo de comprensión más profunda lograda por este enfoque, sino a la idea de capas sucesivas de representaciones. El número de capas que contribuyen a un modelo de datos se denomina profundidad del modelo [20].

El aprendizaje profundo descubre estructuras intrincadas en grandes conjuntos de datos mediante el uso del algoritmo de backpropagation para indicar cómo una máquina debe cambiar sus parámetros internos que se usan para calcular la repre-

sentación en cada capa a partir de la representación en la capa anterior. Las redes neuronales profundas han generado avances en el procesamiento de imágenes, video, voz y audio, mientras que las redes recurrentes han arrojado luz sobre datos secuenciales como texto y voz [23].

3.3.1. Aprendizaje por Transferencia

En la rama de Aprendizaje Automático, existen ciertas técnicas que nos permiten utilizar una arquitectura profunda, como redes neuronales o transformers, ya entrenada previamente con cierto conjunto de datos, con la finalidad de volverla a entrenar con el conjunto de datos de nuestra elección en tareas similares, y evitarnos costo computacional.

Por ejemplo, supongamos que existe el acceso a una Red Neuronal que fue entrenada para clasificar imágenes en 100 categorías diferentes, incluyendo animales, plantas, vehículos y objetos cotidianos. Ahora se quiere entrenar una DNN para clasificar tipos específicos de vehículos. Estas tareas son muy similares, incluso se solapan parcialmente, por lo que se debería intentar reutilizar partes de la primera red [24].

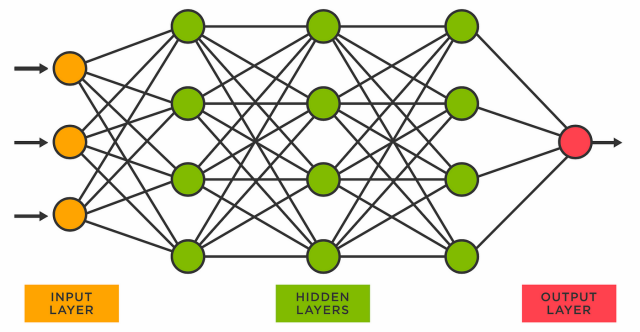

Figura 3.1: Arquitectura de una red neuronal totalmente conectada [1]

3.3.2. Redes Neuronales Artificiales

Las Redes Neuronales, también conocidas como Redes Neuronales Artificiales (RNA) o redes neuronales simuladas (SNN), son un subconjunto del aprendizaje automático y están en el centro de los algoritmos de aprendizaje profundo. Su nombre y estructura se inspiran en el cerebro humano, imitando la forma en que las neuronas biológicas se comunican entre sí.

La neurona es una célula que se encuentra sobre todo en las cortezas cerebrales, compuesta por un cuerpo celular que contiene el núcleo y la mayoría de los componentes complejos de la célula, y muchas prolongaciones ramificadas llamadas dendritas, además de una prolongación muy larga llamada axón. La longitud del axón puede ser unas pocas veces mayor que la del cuerpo celular o hasta decenas de miles de veces mayor.

Cerca de su extremo, el axón se bifurca en muchas ramas llamadas telodendritas, y en la punta de estas ramas hay unas estructuras minúsculas llamadas terminales sinápticas, que están conectadas a las dendritas (o directamente al cuerpo celular) de otras neuronas. A través de estas sinapsis, las neuronas biológicas reciben de otras neuronas breves impulsos eléctricos denominados señales. Cuando una neurona recibe un número suficiente de señales de otras neuronas en unos pocos milisegundos, dispara sus propias señales [24].

Este mecanismo biológico se simula en redes neuronales artificiales, que contienen unidades de cálculo denominadas neuronas. Las unidades computacionales están conectadas entre sí mediante pesos, que desempeñan la misma función que la fuerza de las conexiones sinápticas en los organismos biológicos.

Cada entrada a una neurona se escala con un peso, que afecta a la función calculada en esa unidad. Una red neuronal artificial calcula una función de las entradas propagando los valores calculados desde las neuronas de entrada a la(s) neurona(s) de salida y utilizando los pesos como parámetros intermedios [25]. El aprendizaje se

produce cambiando los pesos que conectan las neuronas.

Al igual que en los organismos biológicos se necesitan estímulos externos para aprender, en las redes neuronales artificiales el estímulo externo lo proporcionan los datos de entrenamiento que contienen ejemplos de pares de entrada-salida de la función que se desea aprender.

Estos pares de datos de entrenamiento se introducen en la red neuronal utilizando las representaciones de entrada para hacer predicciones sobre las etiquetas de salida. Los datos de entrenamiento proporcionan información sobre la corrección de los pesos de la red neuronal en función de la concordancia entre la salida prevista para una entrada concreta y la etiqueta de salida anotada en los datos de entrenamiento. Los errores cometidos por la red neuronal en el cálculo de una función pueden considerarse como una especie de retroalimentación desagradable en un organismo biológico, que conduce a un ajuste de las fuerzas sinápticas.

Del mismo modo, los pesos entre neuronas se ajustan en una red neuronal en respuesta a los errores de predicción. El objetivo de cambiar los pesos es modificar la función calculada para que las predicciones sean más correctas en futuras iteraciones. Por lo tanto, los pesos se cambian cuidadosamente de forma matemáticamente justificada para reducir el error de cálculo en ese ejemplo.

Ajustando sucesivamente los pesos entre neuronas a lo largo de muchos pares de entrada-salida, la función computada por la red neuronal se va refinando con el tiempo, de modo que proporciona predicciones más precisas. Esta capacidad de calcular con precisión funciones de entradas desconocidas mediante el entrenamiento con un conjunto finito de pares de entrada-salida, se denomina generalización del modelo.

La principal utilidad de todos los modelos de aprendizaje automático reside en su capacidad para generalizar su aprendizaje a partir de datos de entrenamiento observados a ejemplos no observados [25].

3.4. Visión por Computadora

Visión por Computadora se refiere a la extracción automatizada de información a partir de imágenes. La información puede ser cualquier cosa, desde modelos 3D, posición de la cámara, detección y reconocimiento de objetos hasta agrupación y búsqueda de contenido de imágenes. Visión por Computadora intenta imitar la visión humana, a veces utiliza un enfoque estadístico y de datos, y a veces la geometría es la clave para resolver problemas [26].

Los problemas de Visión por Computadora han sido de los trabajos más predominantes en el área de aprendizaje profundo desde inicios de la década de 2010, utilizando para ello redes neuronales convolucionales.

Estos modelos también están en el centro de la investigación de vanguardia en conducción autónoma, robótica, diagnóstico médico asistido por IA, sistemas autónomos de caja e incluso agricultura autónoma [20]. Después del éxito obtenido en el área de Procesamiento de Lenguaje Natural con los Transformers, que utilizan mecanismos de auto-atención [2], se ha optado por elegir arquitecturas que utilicen cabezales de atención para el procesamiento de imágenes, como es el Vision Transformer (ViT) [5], el cual, para conjuntos de datos con un volumen extenso de imágenes, se tuvieron resultados competitivos con el estado del arte. En la figura 3.5 se muestra la arquitectura del Vision Transformer.

3.5. Procesamiento de Lenguaje Natural

El lenguaje natural se puede definir como la forma en la que los seres humanos se pueden comunicar entre ellos, ya sea de forma coloquial o de una manera más propia. Esto contrasta con los lenguajes usados en programación o con las notaciones matemáticas que usan reglas rigurosas para su uso. Existe variedad de lenguajes, como el español, inglés, chino mandarín, etc.

Una de las tareas que se tienen para el área de la computación, principalmente con ayuda de la inteligencia artificial, es que las máquinas entiendan el lenguaje de los humanos y se traten de comunicar como nosotros; tenemos como ejemplos actuales los traductores, buscadores en línea, ChatGPT [8]; a esto se le conoce como Procesamiento de Lenguaje Natural. En palabras formales, el NLP (Natural Language Processing) es una subrama de la inteligencia artificial que se encarga del análisis y estudio de la lengua natural con aplicaciones computacionales. Las personas involucradas en estas investigaciones son llamadas "lingüistas computacionales"[27].

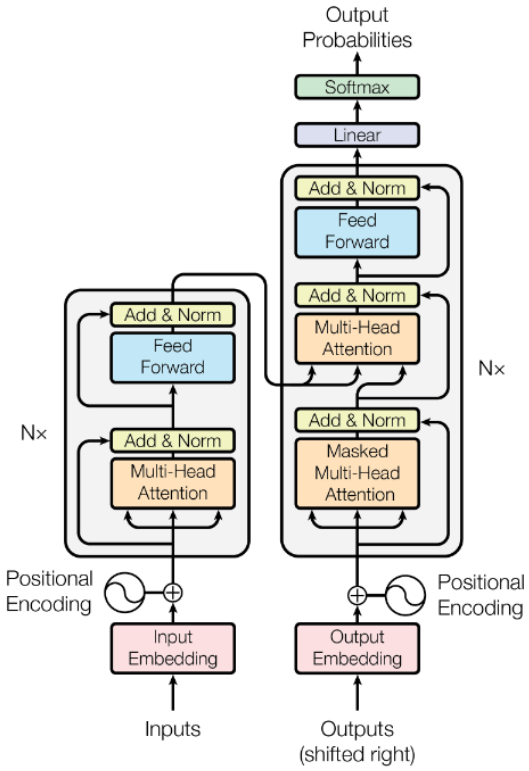

Figura 3.2: Arquitectura de Transformer [2]

3.5.1. Transformer

Transformers

Los Transformers son un tipo de arquitectura de red neuronal que se introdujo en el artículo de Vaswani et al. en 2017 [2]. Se utiliza principalmente para tareas de procesamiento del lenguaje natural, como la traducción de idiomas o la generación de textos. En la figura 3.2 se muestra la arquitectura

El Transformer no utiliza Redes Neuronales Recurrentes (RNN) como los modelos lingüísticos anteriores. Utiliza una técnica llamada “Self Attention", que permite al modelo centrarse en distintas partes de la secuencia al hacer predicciones.

Una función de atención puede describirse como la asignación de una consulta (Query) y un conjunto de pares clave-valor a una salida, donde la consulta, las claves (Keys), los valores (Values) y la salida son todos vectores. La salida se calcula como una suma ponderada de los valores, donde la ponderación asignada a cada valor se calcula mediante una función de compatibilidad de la consulta con la clave correspondiente.

La operación usada para el mecanismo de atención consiste en un producto punto escalado en el cual, la entrada consta de consultas y claves de dimensión d_k, y valores de dimensión d_v . Se calculan los productos escalares de la consulta con todas las claves, se divide cada uno por $\sqrt{d_k}$ y se aplica una función softmax para obtener los pesos de los valores. En la práctica, se calculan la función de atención sobre un conjunto de consultas simultáneamente, agrupadas en una matriz Q. Las claves y los valores también se agrupan en matrices K y V . La matriz de salida se calcula como:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.1}$$

El mecanismo de atención permite a los Transformers tener una memoria a muy largo plazo.

Un modelo de Transformer puede atender o centrarse en todos los tokens anteriores que se hayan generado. Las Redes Neuronales Recurrentes (RNN) también son capaces de fijarse en entradas anteriores. Pero la ventaja del mecanismo de atención es que no sufre las consecuencias de la memoria a corto plazo. Las RNN tienen una ventana de referencia más corta, así que cuando la historia se alarga, las RNN no pueden acceder a palabras generadas antes en la secuencia.

Los Transformers se dice que tienen una arquitectura basada en encoder-decoder. A grandes rasgos, el codificador (encoder) transforma una secuencia de entrada en una representación continua abstracta que contiene toda la información aprendida de esa entrada. A continuación, el decodificador (decoder) toma esa representación continua y, paso a paso, genera una única salida mientras se alimenta de la salida anterior.

La atención multicabezal (Multi-head attention) es un ajuste adicional al mecanismo de auto-atención. El apodo “multicabezal"se refiere al hecho de que el espacio de salida de la capa de autoatención se divide en un conjunto de subespacios independientes, aprendidos por separado: la consulta inicial Q, la clave K y el valor V se envían a través de tres conjuntos independientes de proyecciones densas, lo que da como resultado tres vectores separados. Cada vector se procesa mediante atención neuronal y las tres salidas se concatenan en una única secuencia de salida. Cada uno de estos subespacios se denomina cabeza.

La presencia de las proyecciones densas aprendibles permite que la capa aprenda realmente algo, en lugar de ser una transformación puramente sin estado que requeriría capas adicionales antes o después de ella para ser útil. Además, tener cabezas independientes ayuda a la capa a aprender diferentes grupos de características para cada token, donde las características dentro de un grupo están correlacionadas entre sí, pero son en su mayoría independientes de las características en un grupo diferente [20]. En la figura 3.3 se muestra el proceso detrás del multicabezal de aten-

ción.

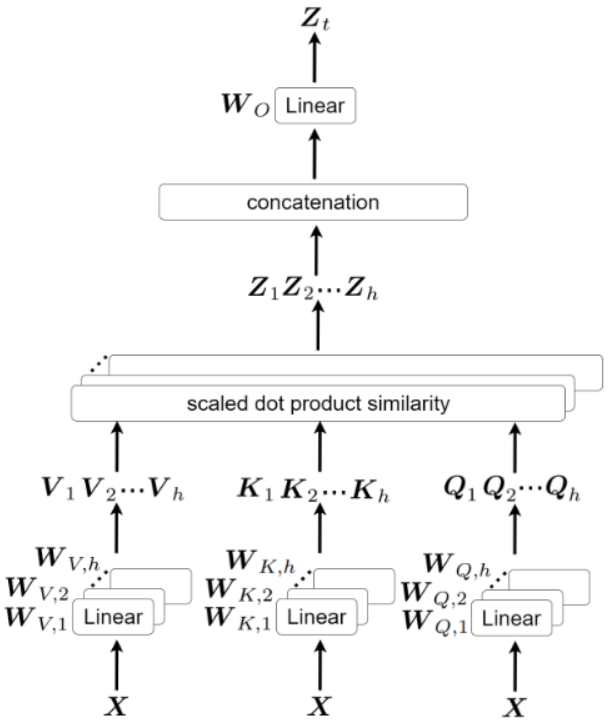

Figura 3.3: multicabezal de atención con *h* cabezales [3]

En otras palabras, el Transformer toma como entrada una secuencia de palabras y primero convierte cada palabra en un vector de números. A continuación, estos vectores pasan por varias capas de Redes Neuronales de auto-atención y feedforward. En las capas de auto-atención, el modelo calcula una suma ponderada de los vectores de entrada, donde las ponderaciones vienen determinadas por la similitud entre cada vector y todos los demás vectores de la secuencia. Esto permite al modelo prestar más atención a determinadas palabras de la secuencia que son más importantes para predecir el resultado.

Las Redes Neuronales feedforward del Transformer se utilizan para procesar la salida de las capas de auto-atención y realizar la predicción final.

Encoder Classifier

Como se mencionaba en la sección anterior, el Transformer original se basa en la arquitectura encoder-decoder que se utiliza ampliamente para tareas como la traducción automática, en la que se traduce una secuencia de palabras de un idioma a otro. En esta sección se describirá más a fondo la función del encoder del transformer, y como este puede ayudar a tareas de clasificación, la cual es parte fundamental para el objetivo de esta investigación.

La función principal del Encoder es codificar la información de la secuencia de entrada en una representación numérica que a menudo se denomina último estado oculto (last hidden state). Aunque elegante en su simplicidad, una debilidad de esta arquitectura es que el estado oculto final del codificador crea un cuello de botella de información: tiene que representar el significado de toda la secuencia de entrada porque esto es todo a lo que tiene acceso el decodificador al generar la salida. Esto es especialmente difícil para secuencias largas, donde la información al principio de la secuencia podría perderse en el proceso de comprimir todo en una única representación fija. La idea principal detrás de la atención es que, en lugar de producir un único estado oculto para la secuencia de entrada, el codificador genera un estado oculto en cada paso al que el decodificador puede acceder. Los modelos de transformers basados solo en el encoder convierten una secuencia de texto de entrada en una representación numérica rica que resulta muy adecuada para tareas como la clasificación de texto o el reconocimiento de entidades nombradas. BERT y sus variantes, como RoBERTa y DistilBERT, pertenecen a esta clase de arquitecturas. La representación calculada para un token determinado en esta arquitectura depende tanto del contexto izquierdo (antes del token) como del derecho (después del token). Esto se denomina a menudo atención bidireccional[28].

Layer Normalization

La normalización estandariza cada entrada sumada utilizando su media y su desviación estándar en los datos de entrenamiento. Las redes neuronales feedforward entrenadas utilizando la normalización por lotes (batch) convergen más rápidamente, incluso con un Stochastic Gradient Descent simple. Además de la mejora en el tiempo de entrenamiento, la estocasticidad de las estadísticas por lotes sirve como regularizador durante el entrenamiento.

A pesar de su simplicidad, la normalización por lotes requiere promedios móviles de las estadísticas de entrada sumadas. En redes feed-forward con profundidad fija, es sencillo almacenar las estadísticas por separado para cada capa oculta. Además, la normalización por lotes no se puede aplicar a tareas de aprendizaje en línea ni a modelos distribuidos extremadamente grandes en los que los minilotes deben ser pequeños.

La normalización de capas o Layer Normalization [29] (LN) es una técnica de aprendizaje profundo que se utiliza para estabilizar el proceso de entrenamiento y mejorar el rendimiento de las redes neuronales. Aborda el problema del cambio de covariables internas (ICS), en el que la distribución de las activaciones dentro de una capa cambia durante el entrenamiento, lo que dificulta que la red aprenda de forma eficaz.

Esta técnica funciona normalizando las activaciones de cada capa de forma independiente en todas las características. Esto significa que la media y la varianza de las activaciones se calculan por separado para cada capa y, a continuación, las activaciones se escalan y se desplazan para obtener una distribución normal estándar (media de 0 y varianza de 1).

Para cada entrada, denotada por **x**, la normalización por capas se puede calcular la siguiente ecuación:

$$y = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \odot \gamma + \beta \tag{3.2}$$

donde μ representa el promedia de las D dimensines, σ^2 se refiere a la varianza de las D dimensiones, ϵ es un valor pequeño que nos ayuda cuando la varianza es pequeña, mientras γ y β son parámetros aprendibles.

Attention Pooling

Una estrategia de pooling se centra en obtener una incrustación (embedding) de tamaño fijo a partir de los estados ocultos del LLM para una secuencia de entrada [30]. El agrupamiento por atención (Attention Pooling) utiliza un mecanismo de autoatención para ponderar la importancia de los diferentes estados ocultos [31]. Attention Pooling suele tomar pesos de atención o los vectores de contexto resultantes (de los mecanismos de atención) y calcula una representación agrupada que captura las señales más relevantes entre tokens, pasos de tiempo o modalidades.

Modelos de Lenguaje Largos Multimodales (MLLMs)

En los seres humanos, un mecanismo fundamental de nuestra percepción sensorial es la capacidad de aprovechar múltiples modalidades de datos de percepción de forma colectiva para relacionarnos adecuadamente con el mundo en circunstancias dinámicas no limitadas, sirviendo cada modalidad como una fuente de información distinta caracterizada por propiedades estadísticas diferentes. Fundamentalmente, un sistema de IA multimodal necesita ingerir, interpretar y razonar sobre fuentes de información multimodales para conseguir capacidades de percepción similares a las humanas.

El aprendizaje multimodal (MML) es un enfoque general para crear modelos de inteligencia artificial capaces de extraer y relacionar información a partir de datos multimodales [32].

Los modelos de lenguaje largos tradicionales se entrenan y aplican principalmente a los datos de texto, pero tienen limitaciones para comprender otros tipos de datos,

Los LLMs puramente textuales, destacan en tareas de generación y codificación de texto, pero carecen de compresión y procesamiento de otro tipo de datos.
Para resolver este problema, los LLM multimodales integran múltiples tipos de datos, superando las limitaciones de los modelos puramente textuales y abriendo posibilidades para manejar diversos tipos de datos. Puede aceptar entradas tanto en forma de imágenes como de texto, y demuestra un rendimiento de nivel humano en varias pruebas de referencia. La percepción multimodal es un componente fundamental para lograr una inteligencia artificial general, ya que es crucial para la adquisición de conocimientos y la interacción con el mundo real [33].

BERT

BERT (Bidirectional Encoder Representations from Transformers) es una implementación específica de la arquitectura de Transformers para tareas de procesamiento del lenguaje natural. La novedad de este modelo, en su momento fue que en lugar de entrenar las secuencias de palabras en una sola dirección, lo hacía de izquierda a derecha y de derecha a izquierda, así, el modelo obtendría un rango más amplio del contexto de las oraciones. El funcionamiento de BERT a grandes rasgos es al hacer uso de Transformer, un mecanismo de atención que aprende contextualmente las relaciones entre palabras. (un encoder que lee el texto de entrada y un decoder que produce una predicción). En el artículo, los investigadores Devlin et al. [4] detallan una novedosa técnica denominada Masked Language Modeling (MLM) o Modelado Lingüístico Enmascarado, que permite el entrenamiento bidireccional en modelos en los que antes era imposible.
El proceso del modelado enmascarado va de la siguiente manera: antes de alimentar a BERT con la secuencia de palabras, el 15 % de las palabras en cada oración son reemplazadas por el token [MASK]. El modelo así intenta predecir el valor original de las palabras emnmascaradas, basándose en el contexto que le proporcionaron

las otras palabras sin máscara de la oración. Para la predicción de las palabras de salida se requiere: agregar una capa de clasificación arriba de la salida del encoder; multiplicar los vectores de salida por la matriz de embeddings, transformándolos en la dimensión del vocabulario; y para finalizar, se calcula la probabilidad de cada palabra en el vocabulario con una función softmax. La función de pérdida de BERT sólo tiene en cuenta la predicción de los valores enmascarados e ignora la predicción de las palabras no enmascaradas. Como consecuencia, el modelo converge más lentamente que los modelos direccionales, una característica que se ve compensada por su mayor conocimiento del contexto.

En el proceso de entrenamiento de BERT, el modelo recibe pares de oraciones como entradas y aprende a predecir si la segunda oración en el par es subsecuente al documento original, mientras que en el otro 50 % se elige una sentencia aleatoria del corpus como segunda oración. Se supone que la oración aleatoria estará desconectada de la primera.

Para apoyar al modelo a distinguir entre las dos oraciones en el entrenamiento, la entrada es procesada, antes de alimentar al modelo de la siguiente forma: se agrega un token [CLS] al principio de la primera frase y otro token [SEP] al final de las dos frases; a cada token se le añade un embedding de oración que indica la oración A o B; por último se agrega un embedding posicional a cada token para indicar su posición en la secuencia.

Al final, para predecir si la segunda oración esta conectada a la primera se se deberán seguir estos pasos: toda la entrada pasará por el modelo; la salida del token [CLS] se tranforma en un vector de tamaño $2 * 1$ usando una capa de clasificación simple; se calculará la probailidad de ser la oración contigua con una función softmax. Al entrenar el modelo BERT, el modelado enmascarado (MLM) y la predicción de la oración siguiente se entrenan juntas, con el objetivo de minimizar la función de pérdida combinada de las dos estrategias.

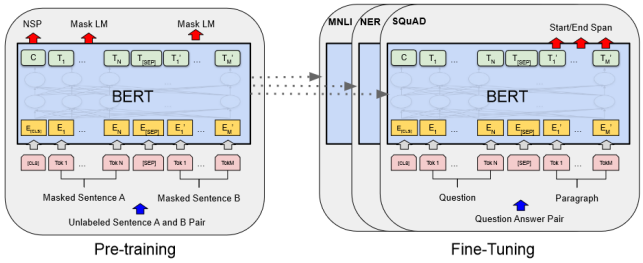

Figura 3.4: Arquitectura de BERT en pre entrenamiento y con ajuste fino de hiperparámetros [4]

La figura 3.4 rescatada de Devlin et al. muestra los procedimentos generales de pre entrenamiento y ajuste fino de BERT.

3.5.2. Vision Transformer

El Visión Transformer, o ViT [5], es un modelo de clasificación de imágenes que emplea una arquitectura similar a la de un Transformer sobre fragmentos de la imagen. Una imagen se divide en fragmentos de tamaño fijo, cada uno de los cuales se incrusta linealmente, se añaden incrustaciones de posición y la secuencia de vectores resultante se alimenta a un codificador Transformador estándar.

Para realizar la clasificación, se utiliza el enfoque estándar de añadir un "token de clasificación" extra aprendible a la secuencia. Vision Transformer (ViT) tiene un proceso de entrada específico para cada imagen en el que la imagen de entrada debe

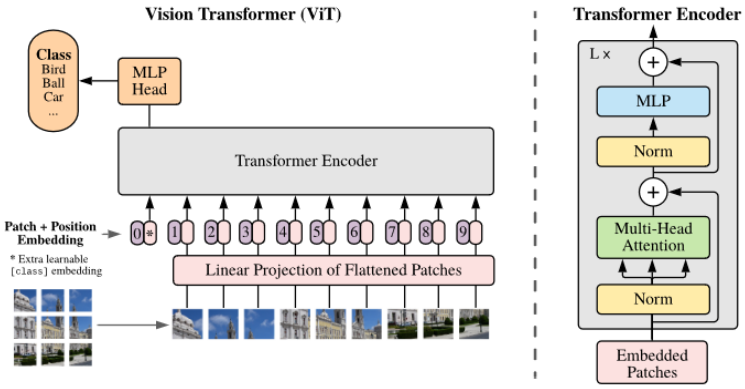

Figura 3.5: Arquitectura de ViT [5]

dividirse en parches de tamaño fijo (por ejemplo, 16 × 16, 32 × 32). Tras pasar por la capa de incrustación lineal y añadir las incrustaciones de posición, todas las secuencias de parches se codifican mediante un codificador Transformer estándar.

Dada una imagen $X \in \mathbb{R}^{HWC}$ (altura H, anchura W, canales C), ViT tiene que remodelar X_p en una secuencia de parches 2D aplanados: $X_p \in \mathbb{R}^{N(P2 \cdot C)}$, donde (H, W) es la resolución de la imagen original, C representa el número de canales, (P, P) es la resolución de cada parche de la imagen, y $N = HW/P^2$ es el número resultante de los parches, que también sirve como longitud de secuencia de entrada efectiva para el Transformer. El Transformer utiliza un vector latente de tamaño constante D en todas sus capas, por lo que se aplanan los parches y se mapea a dimensiones D con una proyección lineal entrenable. Se refieren a la salida de esta proyección como la incrustación de parches.

Similar a al token [class] que se usa en BERT, se antepone un embedding entrenable a la secuencia de parches de embeddings ($z_0^0 = X_{class}$), cuyo estado a la salida del encoder del Transformer (z_L^0) sirve como representación de la imagen y. Tanto durante el pre entrenamiento como durante el ajuste fino se asigna z_L^0 un cabezal de clasificación. El cabezal de clasificación se implementa mediante un MLP con una capa oculta en el preentrenamiento y mediante una sola capa lineal en el ajuste fino.

Los embeddings de posición se añaden a los embededings de parches para conservar la información posicional. Se utilizan embeddings de posición en uni dimensionales entrenables estándar. La secuencia resultante de vectores de incrustación sirve de entrada al encoder.

El encoder del transformer consiste en capas alternas de auto atención multicabezal y bloques de MLP. Una capa de normalización se aplica antes de cada bloques y conexiones residuales despúes de cada uno de estos.

Aumento de Datos

El aumento de datos (Data Augmentation) se refiere a una estrategía para incrementar los datos en el proceso de entrenamiento sin recopilar más ejemplos, con la finalidad de asegurar un buen rendimiento en los modelos de aprendizaje máquina. Los procesos de recopilación y anotación de datos suelen realizarse manualmente y consumen mucho tiempo y recursos. La calidad y representatividad de los datos seleccionados para una tarea determinada suelen depender de la disponibilidad natural de datos limpios en el ámbito concreto, así como del nivel de experiencia de los desarrolladores implicados. En muchos entornos de aplicación del mundo real, a menudo no es posible obtener datos de entrenamiento suficientes. Actualmente, el aumento de datos es la forma más eficaz de solucionar este problema [34]. El aumento de datos se popularizó en las tareas de visión por computadora, mostrando eficacia con transformaciones simples en la imagen, como giro horizontal, aumento en espacios de color y recorte aleatorio. En Procesamiento de Lenguaje Natural, donde el espacio de entrada es discreto, no resulta tan obvio como generar ejemplos aumentados eficaces que capturen las invariantes deseadas.

Con el crecimiento de los Modelos Largos de Lenguaje se ha demostrado una notable comprensión y generación del lenguaje. Sin embargo, el entrenamiento y la optimización de los Modelos de Lenguaje Preentrenados (PLM) requieren enormes cantidades de datos de alta calidad. La mala calidad de los datos y la escasez de los mismos dificultan la mejora de los PLM. Para resolver estos retos, el aumento de datos es una técnica eficaz para generar más datos de entrenamiento disponibles mediante la transformación y la ampliación de los datos existentes, lo que mejora el rendimiento de los modelos en muchas tareas de Procesamiento del Lenguaje Natural (NLP). Los primeros métodos populares de aumento de datos incluyen la sustitución de sinónimos, la reorganización del orden de las palabras y la eliminación aleatoria de palabras.

Con el creciente desarrollo de la ingeniería de prompts y su aplicación en los LLM, muchos investigadores aumentan los datos de entrenamiento iniciales mediante el diseño de prompts elaborados para que los LLM generen conjuntos de datos diversos, lo que contribuye a producir datos de alta calidad y semánticamente ricos [35]. Una de las técnicas usadas con los modelos largos de lenguaje para el aumento de datos es la paráfrasis. Se refiere a textos que se han reescrito o reformulado usando diferentes palabras, formas o estructuras, sin perder el contexto original. La paráfrasis es un método muy eficaz y valioso para el aumento de datos en problemas de PLN, ya que mantiene la fidelidad semántica al tiempo que introduce diversidad léxica. Gracias a esta mejora, el modelo puede aprender nuevas relaciones y ampliar su espacio vectorial [36].

Back Translation

Cuando se utiliza como método de ampliación, la retraducción (Back Translation) se refiere al procedimiento de traducir un ejemplo existente x en el idioma A a otro idioma B y luego volver a traducirlo a A para obtener un ejemplo ampliado $\hat{x}$ [37]. La re traducción puede generar diversas parafrásis mientras se conserva la semántica de la oración original.

MarianMT

MarianMT es un modelo de traducción automática entrenado con el marco Marian [38], escrito en C++ puro. El marco incluye su propio motor de autodiferenciación personalizado y meta-algoritmos eficientes para entrenar modelos codificador-decodificador como BART.

El kit de herramientas Marian se utiliza para implementar un modelo secuencia a secuencia con una arquitectura de tipo Transformer. La arquitectura del modelo consiste en:

- Modelo sequence-to-sequence con una sola capa de RNN tanto en el codificador como en el decodificador.
- RNN bi direccional en el codificador.
- Bloque apilado de GRUs en el codificador y decodificador.
- Mecanismo de atención entre el primer y segundo bloque del decodificador.
- Tamaño de embeddings de 512, tamaño de estado de RNN de 1024.
- Capa de normalización y dropout variacional dentro de bloques GRU y atención.

RandAugment

RandAugment es un método de aumento de datos estocástico para visión y fue propuesta en "RandAugment: Practical automated data augmentation with a reduced search space" de Cubuk et al. para los laboratorios de Google [39].
Simplifica el proceso de aplicar transformaciones aleatorias a las imágenes de entrenamiento, lo que ayuda a los modelos a generalizar mejor al exponerlos a una mayor variedad de variaciones de datos. A diferencia de los métodos más antiguos que requieren un ajuste complejo de las políticas de aumento, RandAugment reduce el espacio de búsqueda a dos hiperparámetros, lo que facilita su implementación y escalado en todos los conjuntos de datos. Los enfoques anteriores, como AutoAugment [40], utilizaban el aprendizaje por refuerzo para encontrar políticas óptimas, lo que resultaba muy costoso desde el punto de vista computacional. RandAugment elimina esta sobrecarga basándose en la aleatoriedad y en valores de magnitud compartidos, al tiempo que sigue obteniendo resultados comparables o mejores.

In-Context Learning

Debido al escalamiento de los modelos largos de lenguaje, tanto de arquitectura como de datos para entrenarse, se demostró que con una nueva técnica, llamada Aprendizaje por contexto (In-Context Learning), estos modelos, tienen la habilidad de aprender por analogías o con pocos ejemplos del contexto.
En los modelos lingüísticos modernos, los tokens que aparecen más tarde en el contexto son más fáciles de predecir que los que aparecen antes. A medida que el contexto se alarga, la pérdida disminuye. En cierto sentido, esto es justo para lo que está diseñado un modelo secuencial (utilizar los elementos anteriores de la secuencia para predecir los posteriores), pero a medida que mejora la capacidad de predecir los tokens posteriores a partir de los anteriores, puede utilizarse cada vez más de formas interesantes (como especificar tareas, dar instrucciones o pedir al modelo que coincida con un patrón) que sugieren que puede considerarse útilmente como un fenómeno propio [41].
Esta nueva técnica se ha visto utilizada en visión por computadora para producir imágenes de salida dado un texto en específico, principalmente.

Drop Out por Modalidades

Se han propuesto varios métodos, entre ellos el conocido Drop Out, para mejorar el proceso de entrenamiento de las redes profundas (DNN), lo que evita el sobreajuste de las DNN. El sobreajuste del modelo se vuelve más difícil cuando se trata del entrenamiento multimodal. El Drop Out por modalidad (m-drop) está diseñado para eliminar una modalidad de datos durante el entrenamiento de acuerdo con una distribución paramétrica, la cual enmascara aleatoriamente las activaciones de la capa oculta a cero para aumentar la capacidad de generalización del modelo subyacente . Matemáticamente, para una entrada multimodal $X = [X_1, X_2, ...X_{n_m}$

y un vector de máscara para el dropout $r \sim Bernoulli(p_m)$, la entrada modificada para la modalidad *i* es $X_i \leftarrow X_i \cdot r_i$, donde p_m es la probabilidad de Drop Out para cada modalidad [42].

Capítulo 4

Metodología

En la siguiente sección se mostrará a detalle el diseño e implementación del modelo propuesto conforme a los objetivos específicos planteados en el primer capítulo. En la figura 4.3 se muestra el proceso a groso modo de la metodología empleada para esta investigación.

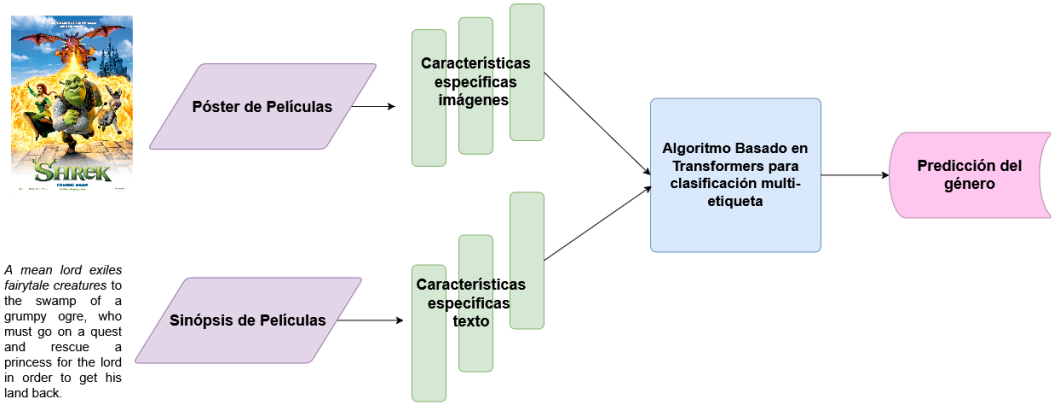

Figura 4.1: Diagrama a bloques de la metodología utilizada

4.1. Obtención y Acondicionamiento del Conjunto de Datos

4.1.1. Selección de la Base de datos

El primer conjunto de datos utilizado para alimentar nuestro modelo propuesto, llamado MM-IMDb, propuesto en el paper "Gated Multimodal Units for Information Fusion"de Arevalo et. al en 2017 [13]. El conjunto de datos MM-IMDb se ha creado a partir de los identificadores de IMDb proporcionados por el conjunto de datos Movielens 20M, que contiene valoraciones de 27.000 películas.

Utilizando la biblioteca IMDbPY, se filtraron las películas que no contenían la imagen de su póster. Como resultado final, el conjunto de datos MM-IMDb comprende 25 959 películas junto con su argumento, póster, géneros y otros 50 campos de metadatos adicionales como año, idioma, guionista, director, relación de aspecto, etc.

Los subconjuntos de entrenamiento, desarrollo y prueba contienen 15.552, 2.608 y 7.799, respectivamente. La muestra se estratificó de modo que los conjuntos de entrenamiento, desarrollo y prueba estuvieran formados por un 60 %, 10 % y 30 % de muestras de cada género, respectivamente.

En total, contamos con 26 etiquetas de géneros de películas: Action, Adult, Adven-

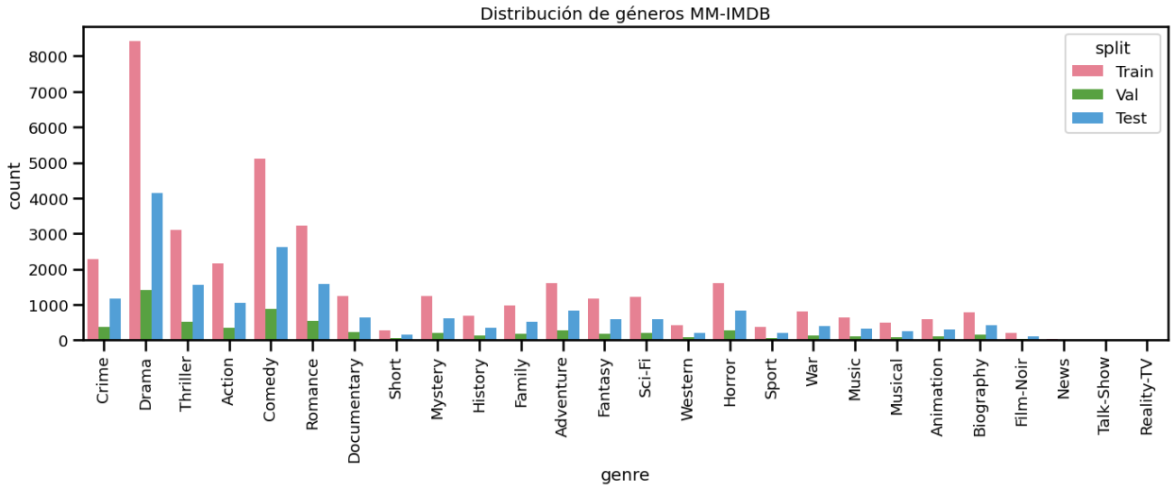

Figura 4.2: Géneros en mm-imdb

ture, Animation, Biography, Comedy, Crime, Documentary, Drama, Familiy, Fantasy, Film-Noir, History, Horror, Music, Musical, Mystery, News, Reality-TV, Romance, Sci-Fi, Short, Sport, Talk-Show, Thriller, War, Western.

El segundo dataset que se utiliza para esta investigación, llamado Moviescope, de 2019 [43], se basa en el conjunto de datos IMDB 5000, que consta de 5043 registros de películas. Este conjunto de datos se publicó bajo una licencia de base de datos abierta como parte de una competición de Kaggle. Se amplió este conjunto de datos recopilando tráilers de vídeos asociados a cada película de YouTube y sinopsis de Wikipedia. Consiste en un dataset multi-etiqueta de 13 etiquetas en total: Mystery, Thriller, Comedy, Action, Crime, Drama, Fantasy, Family, Horror, Biography, Romance, Sci-Fi, Animation.

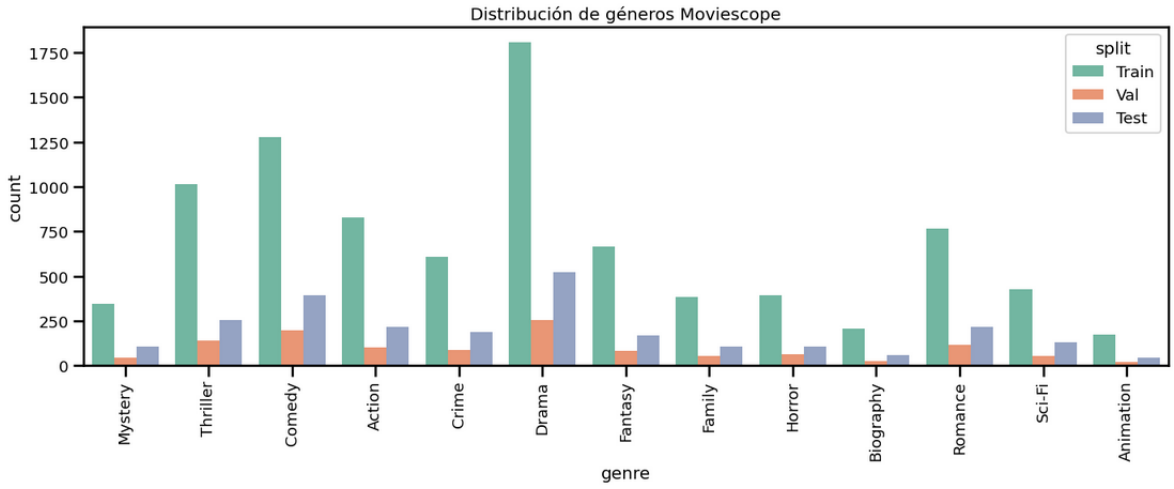

Figura 4.3: Géneros en Moviescope

4.1.2. Preprocesamiento del Dataset

Para la parte visual del modelo, las imágenes tuvieron que cumplir con las siguientes características de pre procesamiento para ser alimentadas por el modelo ViT:

- El parcheado de imágenes es el paso inicial del proceso de Vision Transformer. Consiste en dividir las imágenes en parches más pequeños de un tamaño predeterminado. Por ejemplo, una imagen de 224x224 píxeles puede segmentarse en parches de 16x16 píxeles, lo que da como resultado 196 parches. A continuación, cada parche se aplana en un vector, lo que permite al modelo trabajar con estas piezas más pequeñas y manejables de la imagen.
- Las imágenes se deben de convertir en formato RGB.
- Normalización y estadarización de las imágenes

Todo esto se logra gracias a la clase ViTImageProcessor de HuggingFace.
Para la parte textual, los textos se tokenizan utilizando una versión a nivel de byte de Byte Pair Encoding (BPE) (para caracteres unicode) y un vocabulario de 50.257 caracteres. Las entradas son secuencias de 1024 símbolos consecutivos. Este tokenizador ha sido entrenado para tratar los espacios como partes de los tokens (un poco como sentencepiece) por lo que una palabra se codificará de forma diferente si está al principio de la frase (sin espacio) o no. Solo se eliminaron las etiquets html/xml que podrían causar ruido al momento de utilizaro el procesador textual de HuggingFace para BERT.
Se realizó un aumento de datos en la parte de entrenamiento para el conjunto Moviescope [43], tanto en la parte textual como en las imágenes.
Para el aumento de datos textuales se utilizó la técncia de "back-translation" mediante el modelo MarianMT [38], un modelo de traducción automática neuronal preentrenado disponible en la biblioteca Hugging Face. El procedimiento consiste en traducir el texto original del inglés al francés utilizando el modelo "Helsinki-NLP/opus-mt-en-fr" , y posteriormente volver a traducirlo al inglés mediante el modelo "Helsinki-NLP/opus-mt-fr-en". Este enfoque permite generar múltiples variantes del texto original manteniendo la coherencia semántica, lo que contribuye a

mejorar la generalización del modelo en escenarios con datos limitados.
Para la parte visual se utilizó la técnica RandAugment [39], implementada en PyTorch, que consiste en aplicar de forma aleatoria una secuencia de transformaciones geométricas y fotométricas a cada imagen durante el entrenamiento. En este trabajo se configuró con $numops = 3$, lo que implica que se seleccionan y aplican tres transformaciones aleatorias por imagen, y con $magnitud = 12$, que controla la intensidad de cada transformación. La magnitud define el grado de modificación aplicado en cada operación de aumento de datos. Valores bajos producen transformaciones leves con impacto limitado, mientras que valores altos pueden generar distorsiones excesivas.

4.2. Diseño y entrenamiento del Modelo

Se eligieron los modelos pre entrenados "google/vit-base-patch16-224" y "google-bert/bert-base-uncased" disponibles en la plataforma de Hugging Face como base para el clasificador multi etiqueta. A continuación se mencionarán los detalles del modelo con ajuste fino.

4.2.1. Clasificación Visual

En los primeros experimentos que se realizaron para esta investigación, se utilizó solo el modelo ViT [5] de HuggingFace anteriormente mencionado, para la clasificación del conjunto de datos. Su configuración se refiere a una arquitectura de tamaño base con una resolución de parche de 16x16 y una resolución de ajuste fino de 224x224.
La configuración para este modelo base fue la siguiente:

- La dimensionalidad de las capas encoder y la capa pooler se mantuvo en un tamaño de 768 (hidden state).

- El número de capas ocultas del encoder y los cabezales de atención fueron de 12.
- La función de activación en las capas ocultas que se utilizó fue "GELU".
- tamaño de la imagen en 224 y el tamaño del parche 16. Así mismo el número de canales de la entrada fue de 3.

Los argumentos utilizados para el entrenamiento se ajustaron de la siguiente manera;

- Tasa de aprendizaje de 0.00009.
- Tamaño del lote de entrenamiento y evaluación de 32.
- Se entrenó con un número de 20 épocas.
- decaímiento de los pesos de 0.01

4.2.2. Clasificación Textual

La clasificación de géneros utilizando solo la parte textual se realizó apoyándonos de la arquitectura pre entrenada de BERT [4] que se encuentra igualmente en la plataforma de Hugging Face. La configuración de este modelo corresponde a una arquitectura de 12 cabezales de atención, 12 capas ocultas de encoder y tamaño de la dimensionalidad de las capas de encoder se mantuvo en 768, además usa una función de activación "GELU" y 110 millones de parametros entrenables. Los hiperparámetros para entrenar por ajuste fino fueron los siguientes:

- Tasa de aprendizaje de 0.00002.
- Tamaño de lote en entrenamiento y prueba de 32.
- Número de épocas por entrenamiento de 20.
- 0.01 para decaimiento de los pesos.

4.2.3. ViT-BERT Encoder Classifier

El modelo ViT+BERT Classifier desarrollado para esta investigación está compuesta por dos módulos principales de codificación, diseñados para procesar la información textual y visual, además de una red de clasificación multi etiqueta, que actúa sobre la representación conjunta de las dos modalidades. La implementación se llevó a cabo en PyTorch, y se utilizaron los modelos pre entrenados de Hugging Face, como ya se había mencionado, aprovechando el aprendizaje por transferencia. La combinación del pre entrenamiento de modelos especializados y el uso de mecanismos de atención en el espacio multimodal permite capturar dependencias cruzadas complejas, mejorando la capacidad de clasificación en tareas que involucran datos de distintas modalidades. El módulo de clasificación se integra por varias fases: DropOut por Modalidades, atención cruzada, agrupamiento o pulling por atención y una red totalmente conectada para la clasificación. A continuación se describen a detalle cada sub módulo.

BERTEncoder

Para la representación de palabras, se empleó un codificador basado en BERT, modelo Transformer bi direccional preentrenado. Antes de hacer la codificación, se creó una clase que sirve como "vocabulario", encargada de inicializar los tokens especiales utilizados por BERT ([PAD], [UNK], [CLS], [SEP], [MASK]). Posteriormente, el vocabulario se expande con las palabras presentes en el corpus de entrenamiento. Este proceso asegura que cada palabra o símbolo en las oraciones sea mapeado a un identificador numérico único, garantizando compatibilidad con el modelo BERT y permitiendo manejar palabras fuera del vocabulario mediante el token [UNK].

Los textos se tokenizan empleando el vocabulario previamente construido, mientras que las imágenes se procesan mediante una secuencia de transformaciones (Resize,

CenterCrop, ToTensor, Normalize) compatibles con el modelo ViT. Las etiquetas se convierten en vectores one-hot, permitiendo representar problemas multi etiqueta.
El componente textual se implementó mediante la clase BertEncoder, basada en el modelo preentrenado BERT-base-uncased. Cada secuencia de texto es procesada por BERT para generar representaciones contextuales de sus tokens. Posteriormente, se extrae la representación final de alto nivel de la secuencia, con el fin de obtener un vector de dimensión 768 que representa el significado global de la oración (last hidden state). Este procedimiento permite capturar relaciones sintácticas y semánticas aprendidas por BERT durante su preentrenamiento en grandes corpus textuales. Se congelaron las primeras 6 capas del modelo BERT pre entrenado para reducir el costo computacional y el riesgo de sobre ajuste.

ViTEncoder

Para el procesamiento de imágenes se utilizó el modelo Vision Transformer (ViT), entrenado por Google. ViT adapta la arquitectura transformer al dominio visual mediante el uso de parches (patches) linealmente proyectados, y agrega un token [CLS] al inicio de la secuencia que está diseñado para capturar una representación global de la imagen. El componente visual se implementó mediante la clase ViTImageEncoder, basada en el modelo ViT-base-patch16-224. Cada imagen se divide en parches de 16×16 píxeles, los cuales se tratan como tokens visuales y se procesan por medio de capas de autoatención. De la salida del modelo se obtiene el last hidden state, la representación final de la secuencia de los parches en un vector representativo de dimensión 768. Este vector condensa la información visual relevante de la imagen, permitiendo capturar estructuras espaciales y patrones visuales de alto nivel. Al igual que el modelo de texto, se congelaron las 6 primeras capas del modelo ViT.

Modality Dropout

Se utilizó un mecanismo de *Soft Modality Dropout*, como técnica de regularización para el modelo multimodal. Durante el entrenamiento, se selecciona de manera aleatoria una de las modalidades (texto o imagen) y se atenuan los embeddings por medio de una máscara de Bernoulli, conservando de manera parcial la información. A diferencia del *Hard Modality Dropout*, este enfoque no elimina por completo una de las modalidades, sino que introduce degradación controlada, forzando al modelo a no depender excesivamente de una fuente de información. EL proceso de regularización se realiza de la siguiente manera:
Primero se selecciona aleatoriamente que modalidad se va a degradar. La selección sigue una dsitribución categórica:

$$m \sim Categorical(p_{txt}, p_{img}) \tag{4.1}$$

Esto va a permitir controlar que modalidad se degrada con más frecuencia.
Después se define la probabilidad de conservar una activación como $p_{keep} = 1 - p$, para generar una máscara con distribución estadística de Bernoulli

$$M \sim Bernoulli(p_{keep}) \tag{4.2}$$

y se aplica elemento por elemento sobre los embeddings seleccionados

$$\tilde{X} = X \odot M \tag{4.3}$$

donde X representa los tokens de texto o imagen y $\odot$ el producto elemento por elemento

Cross Attention Fusion

La atención cruzada es un componente clave en las arquitecturas Transformers, donde una secuencia puede atender a la información de otra secuencia [3]. En esta arquitectura en específico, el módulo tiene como objetivo alinear y fusionar información semántica entre las modalidades de texto y visión a nivel de token, permitiendo que cada modalidad incorpore información relevante antes de la agregación global. Teniendo dos representaciones de entrada, X^t como tokens de texto que se obtuvieron de *BERTEncoder*, y X^i como tokens visuales obtenidos por *ViTEncoder*, él módulo aplica dos mecanismos de atención cruzada simétricos:
En el texto atendiendo a la imagen, los tokens de texto actuan como consultas (query), mientras que los tokens de visión actuan como claves y valores (Key and Value) como se muestra en la figura 4.4

$$Attn_{t\leftarrow i} = softmax(\frac{Q_t K_i^T}{\sqrt{D}})V_i \tag{4.4}$$

donde: $Q_t = X^t W_Q^t$, $K_i = X^i W_K^i$, $V = X^i W_V^i$. La salida se combina mediante una conexión residual y normalización:

$$X'^t = LayerNorm(X^t + Attn_{t\leftarrow i}) \tag{4.5}$$

De forma similar, los tokens visuales atienden a los tokens textuales

$$Attn_{i\leftarrow t} = softmax(\frac{Q_i K_t^T}{\sqrt{D}})V_t \tag{4.6}$$

$$X'^i = LayerNorm(X^i + Attn_{i\leftarrow t}) \tag{4.7}$$

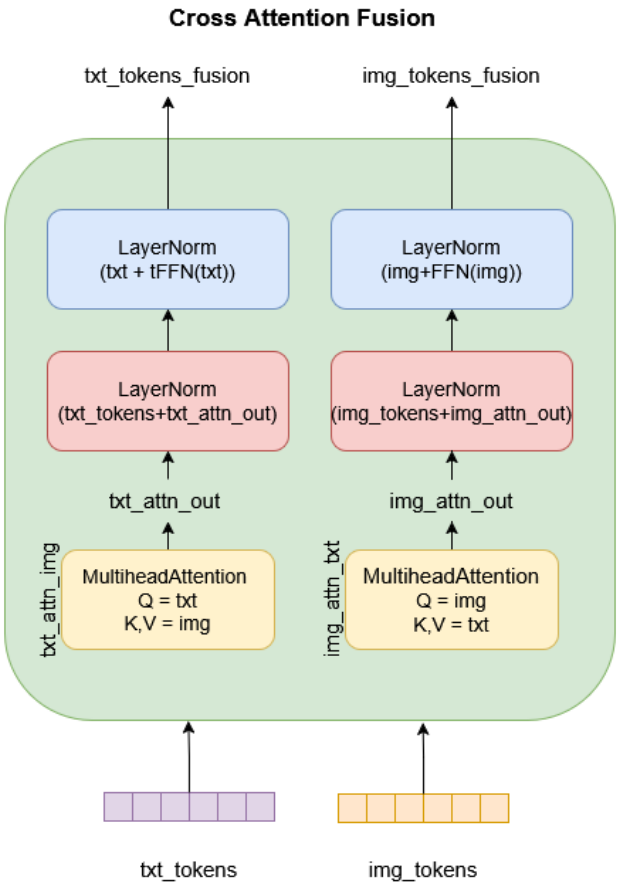

Figura 4.4: Módulo de fusión temprana con Cross Attention

Attention Pooling

Después de pasar los tokens por la atención cruzada (figura 4.4) para fusión temprana de las modalidades, estos mismos pasan por un módulo de *Attention Pooling*, el cual tiene como propósito asignar mayor importancia a los tokens más relevantes para el problema de clasificación, en lugar de usar operaciones como *average pooling* o usar solo el token **CLS**.

Los tokens previamente fusionados con atención cruzada , mostrada en la figura 4.4, X se transforman mediante una capa lineal $h_t = Wx_t + b$, permitiendo mezclar las dimensiones del embedding y aprender las combinaciones relevantes; luego se

aplica una función no lineal $\tilde{h}_t = tanh(h_t)$, introduciendo capacidad no lineal para distinguir tokens relevantes, proyectándose así a un solo vector escalar, $e_t = v^T \tilde{h}_t$, el cual no esta normalizado. Los pesos se logran normalizar mediante una función softmax

$$\alpha_t = \frac{exp(e_t)}{\sum_{k=1}^{T} exp(e_k)} \tag{4.8}$$

La salida se obtiene como una suma ponderada de los tokens

$$z = \sum_{t=1}^{T} \alpha_t x_t \tag{4.9}$$

Esta representación global es el promedio de los tokens, ponderados por su importancia aprendida.

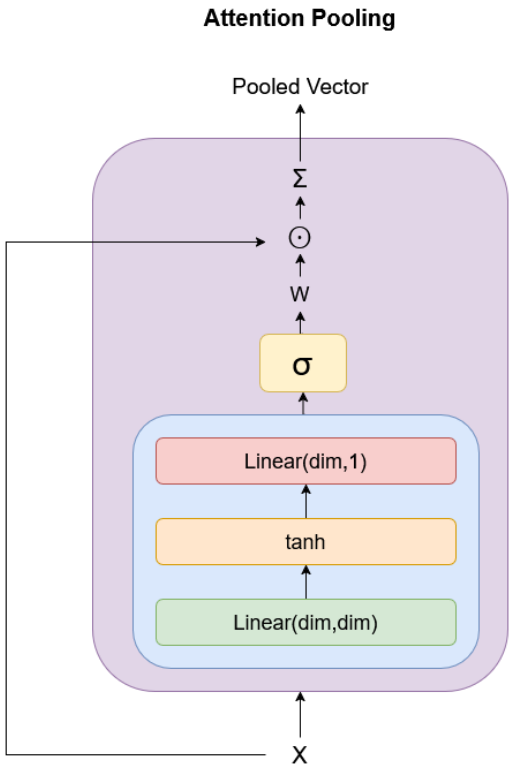

Figura 4.5: Módulo de Attention Pooling

Encoder Layer

Después de utilizar la atención cruzada (figura 4.4) para fusión temprana y attention pooling (figura 4.5) para los tokens de imagen y texto, estos se concatenan y pasan por una capa de codificador de Transformer "*TransformerEncoderLayer*" , mostrada en la figura 4.6. Este módulo termina siendo clave, ya que se termina de refinar la fusión multimodal. Esta arquitectura se conforma de dos sub-bloques: El multicabezal de atención y una red neuronal *feed forward* completamente conectada.

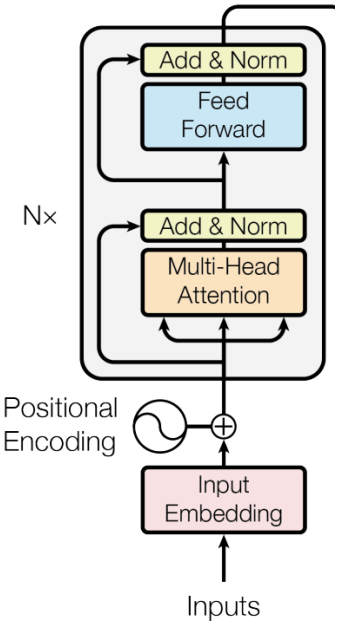

Figura 4.6: Capa Encoder [2]

Dada la entrada, que es el token fusionado z, se tiene la consulta $Q = ZW_Q$, clave $K = ZW_K$, y valor $V = ZW_V$, la atención se calcula como:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4.10}$$

Saliendo del multicabezal de atención , pasa por una normalización por capa, la cual, estabiliza los gradientes y evita la pérdida de información

$$Z' = LayerNorm(Z + Attention(Z)) \tag{4.11}$$

La capa *Feed Forward* es una capa simple compuesta por dos capas lineales, una capa GELU y una capa dropout. En esta capa, los tokens se procesan de forma independiente entre sí y se cree que es la capa donde se produce la mayor parte de la memorización de la información. Nuevamente se presenta otra normalización por

capas *LayerNorm*.

En este caso, la configuración del módulo Encoder se conforma de 8 cabezales de atención, y la dimensionalidad de la *feed forward* es de 1024 unidades. Este Transformer no aprende desde cero, solo refina.

Classifier Neural Network

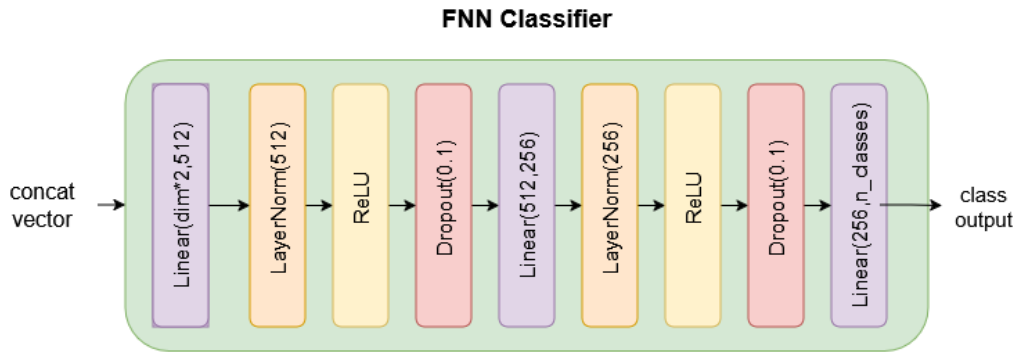

Figura 4.7: Red Neuronal Clasificadora

Este módulo de clasificación, como se muestra en la figura 4.7, se encarga de realizar la predicción final a partir de la representación multimodal, aplicando regularización para mitigar el sobreajuste y mejorar la generalización. La red neuronal clasificadora se conforma de 3 capas lineales; la primera capa reduce la dimensionalidad de la representación multimodal a 512 unidades, así se condensa la información proveniente de texto e imagen a un espacio más compacto. Seguido de esta capa, las unidades pasan por una normalización por capa, mejorando la estabilidad numérica y acelerando la convergencia, se aplica una primera función de activación ReLU, permitiendo al modelo aprender patrones complejos sin incurrir en un costo computacional elevado, posteriormente se aplica un dropout de 0.1, lo cual introduce ruido controlado al entrenamiento.

La segunda capa lineal reduce la dimensionalidad a 256 unidades, actuando como filtro semántico, enfocándose en características más discriminativas para la clasifi-

cación, repitiéndose el bloque de normalización, activación y dropout, reforzando la estabilidad del entrenamiento.

La capa final reduce la dimensionalidad de 256 unidades al número total de clases del problema. Esta capa produce logits independientes para cada etiqueta, adecuado para las clasificaciones multi etiqueta. En esta etapa no se aplica una función softmax, ya que las clases no son mutuamente excluyentes. En su lugar, los logits se transforman posteriormente mediante una función sigmoide y se optimizan utilizando funciones de pérdida como *Binary Cross-Entropy*, o *Focal Loss* en el caso del dataset Moviescope, que utiliza como métrica de evaluación *Mean Average Precision*. Tomando en cuenta las descripciones de los módulos implementados, se presenta a continuación un diagrama a bloques de la Arquitectura propuesta en la figura 4.8

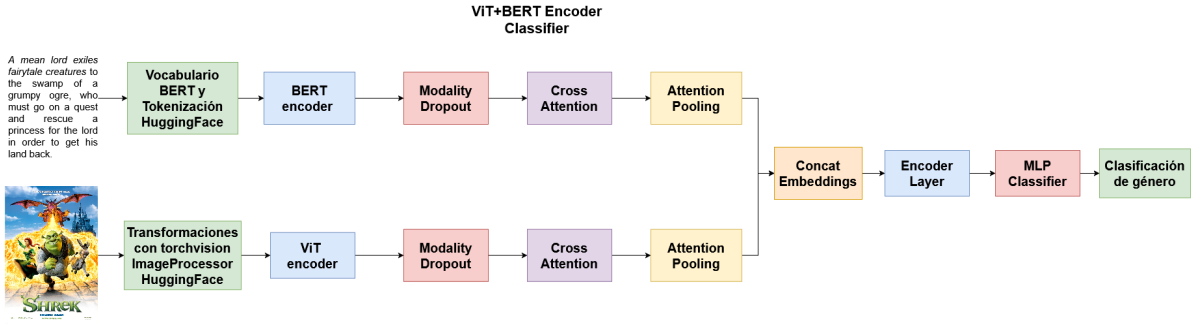

Figura 4.8: Arquitectura ViT+BERT Encoder Classifier

4.3. Funciones de Pérdida

4.3.1. Binary Cross-Entropy

En las tareas de clasificación multi-etiqueta, la pérdida binaria de entropía cruzada (*Binary Cross-Entropy*) desempeña un papel clave al descomponer la clasificación multiclase en múltiples tareas binarias por clase [44]. En la clasificación multietiqueta, cada nodo de la capa de salida predice la probabilidad de que la instancia

pertenezca a una clase concreta. La pérdida de entropía cruzada binaria mide la diferencia entre las probabilidades previstas y las etiquetas de clase reales. La pérdida de entropía cruzada binaria para una sola clase se define de la siguiente manera:

$$Loss = (Y)(-log(Y_{pred})) + (1 - Y)(-log(1 - Y_{pred})) \tag{4.12}$$

donde Y es la etiqueta verdadera (0 ó 1) y Y_{pred} es la probabilidad predicha de la clase.

Para la clasificación multietiqueta con múltiples clases, se calcula la pérdida de entropía cruzada binaria para cada clase y luego se suma para todas las clases a fin de obtener la pérdida total.

4.3.2. Focal Loss

El objetivo inicial de la función de pérdida focal es abordar el problema del equilibrio extremo entre las clases de primer plano y fondo durante el entrenamiento en escenarios de detección de objetos. La pérdida focal se utiliza principalmente para la detección de objetos, pero también demostramos que sus características específicas de dispersión también son aplicables al problema de clasificación con conjuntos de datos desequilibrados [45].

El punto de partida de la pérdida focal es la función de pérdida de entropía cruzada para la clasificación binaria, definida como

$$CE(p, y) = \begin{cases} -log(p), & \text{if } y = 1 \\ -log(1 - p), & \text{otherwise} \end{cases} \tag{4.13}$$

En la que $y \in -1, 1$ denota la etiqueta verdadera para las clases negativas y positivas, respectivamente y $p \in [0, 1]$, indica la probabilidad estimada por el modelo

para la clase con etiqueta $y = 1$. por simplicidad, sea

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \tag{4.14}$$

Con el fin de equilibrar la importancia de las muestras positivas/negativas, se introduce un factor de pesos $\alpha \in [0, 1]$ en una notación similar

$$\alpha_t = \begin{cases} \alpha, & \text{if } y = 1 \\ 1 - \alpha, & \text{otherwise} \end{cases} \tag{4.15}$$

Para reducir la contribución a la pérdida de las muestras fácilmente clasificables, se introduce un factor de modulación *m* con un parámetro de enfoque ajustable $\gamma \geq 0$ en la pérdida de entropía cruzada

$$m = (1 - p_t)^{\gamma} \tag{4.16}$$

Al incorporar estos dos nuevos factores en la ecuación *CE*, la función de pérdida focal propuesta se convierte en

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p_t) \tag{4.17}$$

Tomándo en cuenta que α y γ son dos parámetros que indican la sensibilidad de las muestras fácilmente clasificables.

4.4. Métricas de Evaluación

4.4.1. F1-Score

Se utilizaron las métricas de F1-Score para evaluar el rendimiento de los modelos y compararlos con el estado del arte actual.

La métrica de exactitud (accuracy) calcula cuántas veces un modelo ha realizado una predicción correcta en todo el conjunto de datos. Esta métrica sólo puede ser fiable si el conjunto de datos está equilibrado en cuanto a clases, es decir, si cada clase del conjunto de datos tiene el mismo número de muestras.

F1-Score 4.20 representa la media armónica de precisión 4.18 y exhaustividad 4.19 [24]. La precisión mide cuántas de las predicciones "positivas" realizadas por el modelo eran correctas. La exhaustividad mide cuántas de las muestras de clase positivas presentes en el conjunto de datos fueron identificadas correctamente por el modelo.

$$Precision = \frac{TP}{TP + FP} \tag{4.18}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.19}$$

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{4.20}$$

4.4.2. Average Precision (AP)

La métrica Average Precision fue utilizada para evaluar el conjunto de Datos Moviescope. Esta medida es mayormente utilizada para la detección de objetos en imágenes. La media de los valores de precisión media (AP) se calcula sobre los valores de recuperación (recall) de 0 a 1. La fórmula de AP , se basa en las métricas de mátriz de confusión, intersección sobre unión (IoU), Recall y Precisión. Para calcular el Average Precision:

- Se generan las puntuaciones de predición utilizando el modelo

- Se convierten las puntuaciónes de predición en etiqueta de clase.
- Se calculan la matriz de confusión: los valores verdaderos positivos, flasos positivos, verdaderos negativos y falsos negativos.
- Calcular Precision y recall.
- Calcular el área bajo la curva de precision-recall.
- Medir la precisión media.

4.5. Configuración Experimental

Los experimentos de ambos conjuntos de datos se ejecutaron por 100 épocas con una paciencia de 20 para *Early Stopping*. Se experimentó con diferentes configuraciones de tamaño de lote con 4, 6 y 8, tomando como referencia el de mejor rendimiento. Se entrenó con un optimizador Adam con una tasa de aprendizaje de 0.00005 y un decaimiento de pesos de 0.01.
Se mostró el promedio del rendimiento y la desviación estándar en cinco ejecuciones con diferentes semillas.
Para los experimentos se utilizó una computadora ASUS Phoenix con una tarjeta NVIDIA GeForce RTX 3060 con 12GB GDDR6, ubicada en el laboratorio de Robótica de la Facultad de Ingeniería.

Capítulo 5

Resultados

En esta sección se presentan los resultados obtenidos por la arquitectura propuesta, *ViT+BERT Encoder Classifier*, y se comparan con diversos enfoques del estado del arte, tanto unimodales como multimodales. La evaluación se realiza sobre los conjuntos de datos MM-IMDB y Moviescope, empleando las métricas comúnmente utilizadas en la literatura para cada uno de ellos.

Para MM-IMDB se reportan las métricas micro-F1 ($\mu - F1$), macro-F1 ($m - F1$), weighted-F1 ($W - F1$) y samples-F1 ($s - F1$), mientras que para Moviescope se utilizan micro Average Precision (μAP), macro Average Precision (mAP), weighted AP (WAP) y samples AP (sAP). En todos los casos, los resultados corresponden al promedio y desviación estándar de cinco ejecuciones independientes, cada una con una semilla distinta, con el objetivo de evaluar tanto el rendimiento como la estabilidad del modelo.

De manera general, los resultados muestran que la integración de información textual y visual produce mejoras claras respecto a los modelos unimodales, lo que confirma la pertinencia del enfoque multimodal para la clasificación multietiqueta de géneros cinematográficos. Asimismo, las desviaciones estándar observadas son bajas, lo que sugiere que la arquitectura propuesta mantiene un comportamiento con-

sistente entre ejecuciones.

5.1. Comparación en MM-IMDB

Los resultados para el conjunto de datos MM-IMDB se muestran en la Tabla 5.1. En primer lugar, se observa que los modelos unimodales, BERT y ViT, presentan el rendimiento más bajo, lo cual es esperable dado que únicamente explotan una sola modalidad. En particular, el modelo basado solo en imágenes (ViT) obtiene el peor desempeño, comprobando que la información visual por sí sola resulta insuficiente para capturar adecuadamente el contexto semántico necesario para la clasificación de géneros cinematográficos, reforzando el trabajo de investigación de Monter-Aldana et. al, [18] en el cual muestran un rendimiento de ViT del 32.2 ± 0.6 para $(m - F1)$ y un 43.7 ± 0.3 en $(\mu - F1)$, además de registrar un rendimiento de experimentos con ResNet 152 del 33.3 ± 0.6 y 46.6 ± 0.3 para $(m - F1)$ y $(\mu - F1)$, respectivamente.

Modelo	$\mu - F1$	$m - F1$	$W - F1$	$s - F1$
BERT	63.9	47.1	62.5	64.2
ViT	48.34	28.01	44.9	46.8
ConcatBERT	65.9 ± 0.2	60.5 ± 0.3	-	-
GMU	63.0	54.1	61.7	63
ViLT	64.7	55.3	64.4	64.3
BLIP	67.4	62.8	66.3	67.5
MMBV	66.8 ± 0.4	61.7 ± 0.5	-	-
MMBT	66.6 ± 0.4	61.4 ± 0.5	-	-
BiProjection Multimodal Transformer (SOTA)	68.9 ± 0.1	58.7 ± 0.3	68.8 ± 0.1	69.4 ± 0.2
ViT+BERT Encoder CLF	66.9 ± 0.1	55.2 ± 0.2	66.9 ± 0.2	67.1 ± 0.1

Tabla 5.1: Comparación de la arquitectura propuesta con trabajos anteriores para el dataset MM-IMDB

En contraste, los modelos multimodales muestran mejoras significativas al integrar información textual y visual. La arquitectura propuesta *ViT+BERT Encoder CLF* alcanza un desempeño competitivo, superando a modelos multimodales clásicos como *ConcatBERT*, *MMBV* y *GMU* en varias métricas. En particular, se observa una

mejora consistente en $\mu - F1$, $W - F1$ y $s - F1$, lo que indica una mejor capacidad del modelo para manejar el desbalance de clases y realizar predicciones más robustas a nivel de instancia.

Asimismo, el modelo propuesto supera a enfoques recientes como *ViLT* y *BLIP* en todas las métricas F1, lo que sugiere que la estrategia de fusión mediante atención cruzada, junto con el uso de dropout de modalidad y pooling por atención, contribuye de manera efectiva a una representación multimodal más discriminativa.

No obstante, el modelo *BiProjection Multimodal Transformer*, considerado el estado del arte (SOTA), continúa presentando el mejor rendimiento global en MM-IMDB, superando a la arquitectura propuesta por aproximadamente dos puntos porcentuales en la mayoría de las métricas. Esto indica que, si bien el modelo propuesto es altamente competitivo, aún existe margen de mejora frente a arquitecturas multimodales más complejas.

5.2. Comparación en Moviescope

La tabla 5.2 muestra los resultados obtenidos sobre el conjunto de datos Moviescope. En este escenario, el modelo propuesto *ViT+BERT Encoder CLF* presenta un desempeño competitivo frente a trabajos previos como *MMBT* y *MMBV*, logrando resultados comparables en μAP, mAP y WAP.

Cabe destacar que estos modelos también incorporan técnicas de aumento de datos (como el reemplazo de tokens usando similitud de coseno en embeddings de Word-To-Vec para la parte textual y RandAugmentation para visión) durante el entrenamiento, lo cual refuerza la validez de la comparación.

Es necesario recordar que la arquitectura MMBT, propuesta por el trabajo de Kiela et al. [14] utiliza ResNet 152, en lugar de un codificador ViT para la parte visual del experimento.

Modelo	μAP	mAP	WAP	sAP
MMBT	77.4 ± 0.7	74 ± 0.8	-	-
MMBV	75.5 ± 0.1	79.4 ± 0.3	-	-
BiProjection Multimodal Transformer (SOTA)	81.4 ± 0.6	78.0 ± 0.5	-	87.2 ± 0.4
ViT+BERT Encoder CLF	76.5 ± 0.6	74.8 ± 0.4	78.3 ± 0.4	84.6 ± 0.4

Tabla 5.2: Resutados del modelo propuesto con el dataset Moviescope y comparación con trabajos anteriores

Sin embargo, al igual que en MM-IMDB, el modelo BiProjection Multimodal Transformer mantiene una ventaja significativa en Moviescope. Este mejor desempeño puede atribuirse, en gran medida, a que dicho modelo incorpora una tercera modalidad (audio) durante el entrenamiento, lo cual le permite capturar información adicional que no está disponible para la arquitectura propuesta. Esta diferencia en el número de modalidades explica, en parte, la brecha observada en las métricas de evaluación.

5.3. Efecto del aumento de datos

Como parte del proceso de entrenamiento, se incorporaron estrategias de aumento de datos tanto para texto como para imagen. En la modalidad textual se utilizó *back-translation*, mientras que en la modalidad visual se aplicó *RandAugment*. Estas técnicas buscaron incrementar la diversidad del conjunto de entrenamiento sin alterar significativamente la semántica de las muestras originales.

Aunque en este trabajo no se presenta un estudio de ablación exhaustivo, la inclusión de estas técnicas se consideró relevante para mejorar la capacidad de generalización del modelo, especialmente en un escenario con desbalance entre géneros y alta variabilidad visual entre pósters.

5.4. Discusión general de resultados

En general, los resultados obtenidos permiten extraer varias observaciones relevantes. En primer lugar, se confirma que la modalidad textual aporta una señal semántica más fuerte que la visual cuando se trata de clasificar géneros cinematográficos, lo cual se refleja en el bajo rendimiento de los modelos puramente visuales frente a los modelos basados en texto. No obstante, la incorporación de la modalidad visual sí proporciona información complementaria útil, permitiendo mejorar el rendimiento respecto a enfoques unimodales.

En segundo lugar, la arquitectura propuesta demuestra que una estrategia de fusión basada en encoders preentrenados, atención cruzada, pooling por atención y regularización por modalidad puede ofrecer un desempeño competitivo sin recurrir a estructuras excesivamente complejas. Esto resulta especialmente relevante en contextos donde se busca un balance entre capacidad de representación y simplicidad de implementación.

Finalmente, aunque el modelo no supera al mejor enfoque reportado en la literatura, sí logra posicionarse de manera favorable frente a varios modelos multimodales previos. Esto sugiere que la propuesta constituye una alternativa válida para la clasificación multimodal de películas, y que aún existen oportunidades de mejora, particularmente en el tratamiento de clases minoritarias y en la incorporación de modalidades adicionales.

Capítulo 6

Conclusiones

A partir de los experimentos realizados, se concluye que la arquitectura propuesta, basada en *Vision Transformer* (ViT) para la modalidad visual y *BERT* para la modalidad textual, constituye una alternativa competitiva para la tarea de clasificación multimodal multietiqueta en los conjuntos de datos *MM-IMDB* y *Moviescope*. Los resultados obtenidos muestran que la integración de ambas modalidades permite capturar información complementaria del contenido cinematográfico, logrando un desempeño comparable al de diversos enfoques multimodales reportados en la literatura [18], [15], [46], [13].

Uno de los aportes relevantes de este trabajo es la incorporación de estrategias de aumento de datos específicas para cada modalidad. En la parte textual, se utilizó *back-translation* mediante modelos preentrenados *MarianMT*, con el objetivo de generar variantes semánticamente equivalentes de las sinopsis originales. Esta técnica permitió enriquecer el conjunto de entrenamiento y favorecer la capacidad de generalización del modelo. Por otro lado, en la modalidad visual se empleó *RandAugment*, aplicando transformaciones aleatorias controladas sobre los pósters de películas, lo que contribuyó a robustecer el aprendizaje frente a variaciones visuales comunes como cambios de contraste, rotación, brillo o recorte.

En el conjunto de datos *MM-IMDB*, la arquitectura *ViT+BERT Encoder CLF* alcanzó un valor de $samples\text{-}F1$ de 67.1 ± 0.1, además de mostrar un comportamiento sólido en $\mu\text{-}F1$ y $Weighted\text{-}F1$, lo cual indica una buena capacidad para manejar el desbalance de clases presente en la tarea. Si bien el modelo no supera al estado del arte en todas las métricas, sí logra posicionarse por encima de varios enfoques multimodales previos como *ConcatBERT*, *GMU* y otros modelos base, lo que confirma que la estrategia de fusión propuesta es efectiva.

En el caso de *Moviescope*, el modelo también presentó resultados competitivos, especialmente frente a arquitecturas como *MMBT* y *MMBV*. Aunque el rendimiento obtenido permanece por debajo del modelo *BiProjection Multimodal Transformer*, esta diferencia puede interpretarse en función de la información adicional utilizada por dicho enfoque, ya que incorpora una tercera modalidad (audio), la cual no fue considerada dentro del alcance del presente trabajo. Aun así, los resultados muestran que una arquitectura basada únicamente en texto e imagen puede ofrecer un desempeño sólido en tareas de clasificación multimodal.

Desde el punto de vista metodológico, este trabajo demuestra que no es indispensable recurrir a mecanismos de fusión excesivamente complejos para obtener resultados competitivos. La combinación de encoders preentrenados, atención cruzada, *attention pooling*, congelamiento parcial de capas y una etapa final de clasificación permitió construir una arquitectura funcional, estable y bien adaptada a la tarea. En particular, el uso de atención cruzada permitió que cada modalidad enriqueciera su representación a partir de la otra, mientras que los módulos de *pooling* facilitaron la extracción de información relevante antes de la etapa de clasificación.

Asimismo, el uso de congelamiento parcial en las primeras capas de *BERT* y *ViT* permitió aprovechar el conocimiento previamente aprendido durante el preentrenamiento, concentrando el ajuste en los módulos de fusión y clasificación. Esta decisión metodológica resultó adecuada para mantener un entrenamiento más contro-

lado y orientar el aprendizaje hacia la integración multimodal, en lugar de reajustar completamente los encoders.

Con base en estos hallazgos, se considera que la hipótesis planteada en esta investigación se cumple parcialmente, en el sentido de que un modelo basado en encoders de Transformers, atención cruzada, regularización multimodal y congelamiento de capas puede alcanzar resultados competitivos frente a enfoques multimodales reportados en el estado del arte. Si bien no se supera al mejor modelo reportado, la arquitectura propuesta demuestra que es posible obtener un rendimiento sólido mediante una estrategia de integración multimodal estructuralmente más accesible y reproducible.

6.1. Limitaciones y trabajo a futuro

A pesar de los resultados obtenidos, este trabajo presenta algunas limitaciones. En primer lugar, la arquitectura fue evaluada únicamente sobre dos modalidades (texto e imagen), por lo que no aprovecha otras fuentes de información potencialmente útiles, como audio, subtítulos o metadatos. Además, aunque se incorporaron estrategias de aumento de datos, no se realizó un estudio de ablación que permitiera cuantificar de manera aislada la contribución de cada componente del modelo. Estas limitaciones representan oportunidades claras para profundizar en trabajos posteriores.

Finalmente, este trabajo abre distintas líneas para investigación futura. Entre ellas, destaca la incorporación de modalidades adicionales, como audio o metadatos, la exploración de estrategias de fusión más profundas o jerárquicas, así como el análisis sistemático del impacto individual de cada componente mediante estudios de ablación. También sería valioso evaluar con mayor detalle el efecto aislado de las técnicas de aumento de datos y explorar variantes más recientes de modelos pre-

entrenados para texto e imagen. En conjunto, estos elementos podrían contribuir a mejorar aún más el desempeño del modelo y acercarlo al estado del arte actual.

Referencias

[1] Amzhao, "Quantum Neural Networks - MIT 6.s089—Intro to Quantum Computing - Medium," 2 2023.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[3] B. Ghojogh and A. Ghodsi, "Attention mechanism, transformers, bert, and gpt: Tutorial and survey," 12 2020.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[6] G. Ramirez-Alonso, O. Prieto-Ordaz, R. López-Santillan, and M. Montes-Y-Gómez, "Medical report generation through radiology images: an overview," *IEEE Latin America Transactions*, vol. 20, no. 6, pp. 986–999, 2022.

[7] J. Abraham, R. Ng, M. Morelato, M. Tahtouh, and C. Roux, "Automatically classifying crime scene images using machine learning methodologies," *Forensic Science International: Digital Investigation*, vol. 39, p. 301273, 2021.

[8] OpenAI, "Gpt-4 technical report," 2023.

[9] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021.

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," 2021.

[11] C. Ferragu, P. Chagniot, and V. Coyette, "Multimodal clip inference for meta-few-shot image classification," 2024.

[12] A. C. Li, M. Prabhudesai, S. Duggal, E. Brown, and D. Pathak, "Your diffusion model is secretly a zero-shot classifier," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2206–2217, 2023.

[13] J. Arevalo, T. Solorio, M. M. y Gómez, and F. A. González, "Gated multimodal units for information fusion," *ArXiv*, vol. abs/1702.01992, 2017.

[14] D. Kiela, S. Bhooshan, H. Firooz, and D. Testuggine, "Supervised multimodal bitransformers for classifying images and text," *arXiv preprint arXiv:1909.02950*, 2019.

[15] D. A. Moreno-Galván, R. López-Santillán, L. C. González-Gurrola, M. Montes-Y-Gómez, F. Sánchez-Vega, and A. P. López-Monroy, "Automatic movie genre classification emotion recognition via a biprojection multimodal transformer," *Information Fusion*, vol. 113, p. 102641, 2025.

[16] T. Liang, G. Lin, M. Wan, T. Li, G. Ma, and F. Lv, "Expanding large pre-trained unimodal models with multimodal information injection for image-text multimodal classification," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15471–15480, 2022.

[17] F. Pahde, M. Puscas, T. Klein, and M. Nabi, "Multimodal prototypical networks for few-shot learning," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2643–2652, 2021.

[18] I. Monter-Aldana, A. P. Lopez Monroy, and F. Sanchez-Vega, "Dynamic regularization in UDA for transformers in multimodal classification," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 8700–8711, Association for Computational Linguistics, July 2023.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[20] F. Chollet, *Deep learning with Python*. Simon and Schuster, 2021.

[21] A. C. Müller and S. Guido, *Introduction to machine learning with Python: a guide for data scientists*. .ºReilly Media, Inc.", 2016.

[22] M. W. Berry, A. Mohamed, and B. W. Yap, *Supervised and unsupervised learning for data science*. Springer, 2019.

[23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[24] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. .ºReilly Media, Inc.", 2022.

[25] C. Aggrawal, "Neural networks and deep learning: A textbook," 2018.

[26] J. E. Solem, *Programming Computer Vision with Python: Tools and algorithms for analyzing images*. .ºReilly Media, Inc.", 2012.

[27] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural language processing: python and NLTK*. Packt Publishing Ltd, 2016.

[28] L. Tunstall, L. Von Werra, and T. Wolf, *Natural language processing with transformers*. .ºReilly Media, Inc.", 2022.

[29] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[30] Y. Tang and Y. Yang, "Pooling and attention: What are effective designs for LLM-based embedding models?," 2025.

[31] H. Touvron, M. Cord, A. El-Nouby, P. Bojanowski, A. Joulin, G. Synnaeve, and H. Jégou, "Augmenting convolutional networks with attention-based aggregation," 2021.

[32] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 12113–12132, 2023.

[33] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu, " Multimodal Large Language Models: A Survey ," in *2023 IEEE International Conference on Big Data (BigData)*, (Los Alamitos, CA, USA), pp. 2247–2256, IEEE Computer Society, Dec. 2023.

[34] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, 2022.

[35] Y. Chai, H. Xie, and J. S. Qin, "Text data augmentation for large language models: A comprehensive survey of methods, challenges, and opportunities," 2025.

[36] Radliński, M. Guściora, and J. Kocoń, *Backtranslation and Paraphrasing inthe LLM Era? Comparing Data Augmentation Methods for Emotion Classification*, p. 3–17. Springer Nature Switzerland, 2025.

[37] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "Unsupervised data augmentation for consistency training," *arXiv preprint arXiv:1904.12848*, 2019.

[38] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, and A. Birch, "Marian: Fast neural machine translation in c++," 2018.

[39] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

[40] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," 2019.

[41] C. Olsson, N. Elhage, N. Nanda, N. Joseph, N. DasSarma, T. Henighan, B. Mann, A. Askell, Y. Bai, A. Chen, T. Conerly, D. Drain, D. Ganguli, Z. Hatfield-Dodds, D. Hernandez, S. Johnston, A. Jones, J. Kernion, L. Lovitt, K. Ndousse, D. Amodei, T. Brown, J. Clark, J. Kaplan, S. McCandlish, and C. Olah, "In-context learning and induction heads," *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

[42] C. Wang, M. Niepert, and H. Li, "Lrmm: Learning to recommend with missing modalities," 2018.

[43] P. Cascante-Bonilla, K. Sitaraman, M. Luo, and V. Ordonez, "Moviescope: Large-scale analysis of movies using multiple modalities," *ArXiv*, vol. abs/1908.03180, 2019.

[44] T. Kobayashi, "Two-way multi-label loss," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7476–7485, 2023.

[45] G. S. Tran, T. P. Nghiem, V. T. Nguyen, C. M. Luong, and J.-C. Burie, "Improving accuracy of lung nodule classification using deep learning with focal loss," *Journal of healthcare engineering*, vol. 2019, no. 1, p. 5156416, 2019.

[46] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *International conference on machine learning*, pp. 5583–5594, PMLR, 2021.

EDUCATION

Universidad Autónoma de Chihuahua – Chihuahua, MX **Jan 2024 - Dec 2025**
Master Degree in Computer Engineering | Program funded by SECIHTI.

Relevant Coursework: Algorithms, Computer Vision, Natural Language Processing, Generative AI, Big Data Analysis

Universidad Autónoma de Chihuahua – Chihuahua, MX **Aug 2018 - Dec 2022**
Bachelor Degree in Computer Science and Engineering

Relevant Coursework: Algorithms, Machine Learning, Software Engineering, Platform- based Development, Databases

TECHNICAL SKILLS

Programming: Python, C, C++, JavaScript, Java, SQL, HTML, CSS, Shell, C#, TypeScript
Frameworks: React, Node.js, Flask, TensorFlow, PyTorch, QT5, HuggingFace, Streamlit, Stanza, Numpy, Matplolib, scikit-learn, pandas
Tools: Git, Docker, Linux, Windows, Colaboratory, jupyter Notebook, Tableau, Power BI, Figma, Jira, AWS, Anaconda, Office.
Languages: Spanish (Native), English (Intermediate)

EXPERIENCE

Associate Systems Engineer
AUTOZONE BTSSC | Chihuahua, MX **Dec 2022 – Feb 2024**

- C++ Developer and Researcher within an international development team supporting retail systems in multiple countries.
- Analyzed application and system logs in Linux virtual environments to detect, diagnose, and resolve issues.
- Debugged and corrected software defects, contributing to system reliability and operational efficiency.

Private Tutor
Mundo Newton S.A **Jan 2020 – Dec 2022**

- Private tutoring in high school and college level subjects, such as basic programming, calculus, algebra, numerical methods, oriented object programming and data structures.
- Assisted educators in creating training materials and example programs

PROJECTS

Bachelor Degree Thesis | Author Profiling By Genre, Profession and Political Ideology Using Traditional Machine Learning and Deep Learning

- Implementation of machine learning and deep learning models for the analysis of political ideologies in some specific user profiles in Twitter.

Master Degree Thesis | Classification Of Films By Genre Using A Multimodal Architecture Based On Transformers

- Design and implementation of a multimodal architecture based on transformers for image and text classification, implemented in PyTorch using Hugging Face models. The work included text and image data augmentation (back translation, RandAugment), and analysis of multimodal feature fusion for improved classification performance.

Maritrini Velázquez Ruiz

Computer Science Engineer | mari.trinivel@gmail.com |