Arquitectura Two-Tower Multimodal para Recomendación Musical Secuencial: Fusión de Audio, Texto e Imagen mediante Aprendizaje Contrastivo (InfoNCE)

César Aguirre-Calzadilla[†], Gustavo Hernández-Angeles[†] and Diego Paniagua-Molina[†]

Mathematics Research Center (CIMAT—Centro de Investigación en Matemáticas), Graduate Program in Statistical Computing, Nuevo León, Mexico

Abstract

Este trabajo presenta una arquitectura multimodal tipo *Two-Tower* para la recomendación secuencial de música, diseñada para mitigar el problema de arranque en frío y enriquecer la representación latente de los ítems. El modelo integra cuatro modalidades: audio (espectrogramas Mel procesados con ResNet-18), texto (letras codificadas con mDeBERTa y LoRA), imágenes (carátulas procesadas con ResNet-18 preentrenada) y metadatos tabulares. Utilizando un dataset *ad-hoc* de 10,000 canciones y 3 millones de interacciones de Last.fm, Spotify y YouTube, empleamos una estrategia de fusión tardía y una función de pérdida *Triplet Loss* para optimizar el espacio métrico compartido. Los resultados experimentales muestran un Recall@10 de 0.6225 y un NDCG@10 de 0.5478, superando a los enfoques unimodales y validando la eficacia de la fusión de información heterogénea para capturar la semántica compleja de la experiencia musical.

Keywords

Sistemas de Recomendación, Aprendizaje Multimodal, Two-Tower, Fusión Tardía, Procesamiento de Audio, NLP, Visión por Computadora

1. Introducción

La proliferación de plataformas de *streaming* musical ha transformado radicalmente el acceso a la música, generando catálogos que superan las decenas de millones de pistas. En este contexto de sobrecarga de información, los sistemas de recomendación (RS) se han convertido en herramientas indispensables para filtrar contenido y personalizar la experiencia del usuario. Tradicionalmente, estos sistemas han dependido en gran medida del filtrado colaborativo (CF), que infiere preferencias basándose en patrones de interacción históricos [1]. Sin embargo, los enfoques puramente colaborativos enfrentan limitaciones inherentes, notablemente el problema de arranque en frío (*cold-start*), donde el sistema es incapaz de recomendar ítems nuevos o a usuarios nuevos debido a la ausencia de interacciones previas. Además, la música es una experiencia altamente contextual y subjetiva, donde factores como el estado de ánimo, el género y las características acústicas juegan un papel crucial que el CF puro a menudo ignora.

En la industria actual, la arquitectura predominante para la recuperación eficiente de ítems en catálogos masivos es el enfoque *Two-Tower*, popularizado por YouTube [2] y perfeccionado por Google [3]. Para el modelado de preferencias de usuario, los enfoques secuenciales basados en auto-atención, como SASRec [4], han demostrado superar a los métodos tradicionales, capturando la evolución dinámica de intereses similar a lo observado en plataformas como TikTok o Alibaba [5]. Sin embargo, en el dominio musical, el problema de arranque en frío persiste. Inspirados por los trabajos pioneros de Spotify en el uso de redes convolucionales sobre espectrogramas de audio [6], nuestra propuesta integra estas técnicas en un marco multimodal.

Para mitigar estas limitaciones, la investigación reciente ha girado hacia enfoques híbridos y basados en contenido que explotan las características intrínsecas de los ítems. En el dominio musical, la música es una entidad inherentemente multimodal: se percibe a través de la señal acústica (audio), se interpreta

[†] Estos autores contribuyeron de igual manera.

cesar.aguirre@cimat.mx (C. Aguirre-Calzadilla); gustavo.hernandez@cimat.mx (G. Hernández-Angeles); diego.paniagua@cimat.mx (D. Paniagua-Molina)

semánticamente a través de las letras (texto), se asocia visualmente con el arte del álbum (imagen) y se categoriza mediante metadatos editoriales (tabular). A pesar de esta riqueza, la mayoría de los sistemas de recomendación de música basados en contenido se han centrado predominantemente en una sola modalidad, típicamente el audio o los metadatos, ignorando la complementariedad semántica que ofrecen las otras fuentes de información.

Este trabajo presenta un sistema de recomendación musical multimodal diseñado para capturar y fusionar representaciones latentes de cuatro modalidades distintas: audio, texto, imagen y metadatos tabulares. Nuestra hipótesis central es que la integración de estas fuentes heterogéneas permite construir un espacio de representación más robusto y semánticamente rico, mejorando la precisión de las recomendaciones y mitigando el problema de la escasez de datos de interacción.

1.1. Contribuciones

Las principales contribuciones de este trabajo son las siguientes:

- **Construcción de un Dataset Multimodal**: Desarrollamos un conjunto de datos *ad-hoc* que vincula interacciones de usuarios de Last.fm con señales de audio (YouTube), letras (Genius), imágenes de portadas (Spotify) y metadatos estructurados, creando un recurso unificado para la investigación en recomendación multimodal.
- **Arquitectura Two-Tower Multimodal**: Proponemos una arquitectura de red neuronal de dos torres (*Two-Tower*) que desacopla la codificación del usuario y del ítem. La torre del ítem implementa una estrategia de fusión tardía (*Late Fusion*), integrando embeddings provenientes de codificadores especializados:
  - **Audio**: ResNet-18 modificada para procesar espectrogramas Mel de un solo canal.
  - **Imagen**: ResNet-18 preentrenada en ImageNet para extraer características visuales de las portadas.
  - **Texto**: Modelo de lenguaje mDeBERTa-v3-base optimizado eficientemente mediante Low-Rank Adaptation (LoRA).
  - **Tabular**: Perceptrones multicapa (MLP) para procesar metadatos numéricos y categóricos.
- **Evaluación Integral**: Evaluamos el desempeño del modelo utilizando métricas de ranking estándar (Recall@k, NDCG@k) y demostramos la efectividad del enfoque propuesto en un escenario de recomendación realista, superando a las líneas base unimodales.

1.2. Organización del Documento

El resto de este documento está organizado de la siguiente manera: la Sección 4 revisa la literatura existente sobre sistemas de recomendación y aprendizaje multimodal. La Sección 5 detalla el proceso de recolección y preprocesamiento de datos. La Sección 6 describe la arquitectura del modelo propuesto y la estrategia de entrenamiento. La Sección 7 presenta los resultados experimentales y el análisis de métricas. Finalmente, la Sección 8 ofrece una discusión crítica de los hallazgos y la Sección 9 concluye el trabajo delineando direcciones futuras.

2. Gestión y Planeación

2.1. Matriz RACI

Se asignaron responsabilidades específicas a los integrantes del equipo: César Aguirre (C), Gustavo Hernández (G) y Diego Paniagua (D), para optimizar la colaboración y asegurar el cumplimiento de los objetivos del proyecto. Cada integrante asumió roles de liderazgo en diferentes etapas, manteniendo una comunicación constante. La matriz de asignación de responsabilidades (RACI) se detalla en la Tabla 1.

Table 1
Matriz RACI del Proyecto (R: Responsable, A: Aprobador, C: Consultado, I: Informado)

Actividad	César (C)	Gustavo (G)	Diego (D)
Recolección de Datos (APIs)	R	C	C
Preprocesamiento Multimodal	C	R	I
Diseño de Arquitectura Two-Tower	C	C	R
Implementación del Modelo (InfoNCE)	R	C	I
Entrenamiento y Ajuste	I	R	C
Evaluación de Métricas	C	I	R
Redacción del Reporte Técnico	R/A	R/A	R/A

2.2. Estructura de Desglose del Trabajo (WBS)

- **1. Datos**: extracción (Last.fm, YouTube, Genius, Spotify) y preprocesamiento (Mel-spectrograms, embeddings).
- **2. Modelado**: codificadores unimodales (ResNet, mDeBERTa), arquitectura Two-Tower y Late Fusion.
- **3. Evaluación**: métricas (Recall@k, NDCG@k) y experimentos.
- **4. Entrega**: reporte y código.

2.3. Ruta Crítica

Las tareas críticas fueron: 1) sincronización de datos multimodales, 2) pre-cómputo de características, 3) entrenamiento del modelo Two-Tower con fusión tardía.

2.4. Infraestructura y Flujo de Trabajo

El flujo de trabajo se estructura en tres pilares fundamentales para garantizar la reproducibilidad y la colaboración eficiente: gestión de dependencias, control de versiones de código y control de versiones de datos.

2.4.1. Gestión de Dependencias: uv

Para garantizar la consistencia entre los entornos de desarrollo, se utiliza **uv** como gestor de paquetes. Esta herramienta, escrita en Rust, destaca por su velocidad y permite asegurar que todos los miembros del equipo utilicen exactamente las mismas versiones de las librerías, eliminando conflictos de compatibilidad y gestionando la versión de Python del proyecto automáticamente.

2.4.2. Control de Versiones: Git y GitHub

El manejo del código fuente se realiza mediante una estrategia de ramificación *Gitflow* simplificada, integrando un motor local (Git) y una plataforma en la nube (GitHub).

- **Estrategia de Ramas**: se utiliza `main` para producción, `develop` para integración y ramas `feat/...` para el desarrollo de nuevas características.
- **Alcance**: Git gestiona exclusivamente archivos ligeros de código y configuración (`.py`, `.md`, `.yaml`, `.ipynb`).

2.4.3. Gestión de Datos: DVC

Dado que Git no es adecuado para archivos binarios pesados, se implementó **DVC (Data Version Control)** con almacenamiento remoto en Google Drive.

- **Almacenamiento Híbrido**: Git almacena punteros ligeros (`.dvc`), mientras que los archivos reales (audios, imágenes) residen en la nube.

- **Reproducibilidad**: vincula versiones exactas del código con versiones exactas de los datos, permitiendo replicar experimentos con precisión.

3. Metodología TRIZ

Se aplicó TRIZ para resolver la contradicción entre precisión del modelo y costo computacional.

3.1. Contradicción y Principios

Buscamos aumentar la complejidad semántica (multimodalidad) sin hacer inviable el entrenamiento. Se aplicaron los siguientes principios:

- **Acción Preliminar**: pre-cómputo de espectrogramas y embeddings textuales para reducir carga en tiempo real.
- **Calidad Local**: codificadores especializados (ResNet, mDeBERTa) optimizados para cada modalidad.
- **Fusión**: estrategia de *Late Fusion* para integrar representaciones de alto nivel.
- **Cambio de Parámetros**: uso de LoRA para adaptar modelos de lenguaje grandes con pocos recursos.

Esta metodología validó el diseño de una arquitectura eficiente y potente.

4. Trabajos Relacionados

El desarrollo de sistemas de recomendación efectivos requiere abordar tanto la naturaleza dinámica de las preferencias de los usuarios como la riqueza de información contenida en los ítems. En esta sección, analizamos la literatura existente en torno a dos pilares fundamentales: los modelos secuenciales y el aprendizaje multimodal. Además, contrastamos los enfoques clásicos con las técnicas de aprendizaje profundo y destacamos cómo nuestra propuesta integra estos avances en una arquitectura unificada para superar las limitaciones de los trabajos previos.

4.1. Modelos Secuenciales

Hidasi et al. [7] introdujeron RNNs para recomendaciones basadas en sesiones, capturando dinámicas temporales pero ignorando el contenido, lo que limita su desempeño en *cold-start*.

4.2. Aprendizaje Multimodal

Oramas et al. [8] combinaron texto, audio e imagen mediante fusión temprana, superando modelos unimodales pero sin mecanismos de atención. Won et al. [9] utilizaron aprendizaje métrico con redes siamesas para recuperación basada en etiquetas, destacando la importancia de espacios latentes compartidos.

4.3. Enfoques Clásicos vs. Deep Learning

Murauer and Specht [10] mostraron la eficacia de XGBoost con características artesanales para clasificación de géneros. En contraste, nuestra propuesta utiliza Deep Learning (ResNet-18) para extraer características jerárquicas automáticamente.

4.4. Diferenciación de la Propuesta

A diferencia de los trabajos previos, este proyecto propone una arquitectura *Two-Tower* que:

- Integra cuatro modalidades (audio, texto, imagen, tabular) mediante *Late Fusion*.
- Combina codificación secuencial del usuario con representaciones ricas de contenido.
- Utiliza técnicas eficientes como LoRA para texto.

5. Base de Datos y Recolección

Para este trabajo, se construyó un dataset multimodal *ad-hoc* que integra información de audio, texto, imágenes y metadatos tabulares, alineados con un historial de interacciones de usuarios. El conjunto de datos base consta de 10,000 canciones únicas y más de 3 millones de interacciones provenientes de aproximadamente 850 usuarios de la plataforma Last.fm. A continuación, se detalla el proceso de adquisición y procesamiento para cada modalidad.

5.1. Interacciones y Metadatos (Last.fm y Spotify)

Los datos de interacción usuario-ítem se obtuvieron de Last.fm, recopilando eventos de reproducción que incluyen metadatos del usuario (género, país de residencia) y detalles de la reproducción (tiempo, canción). Complementariamente, se utilizaron identificadores de pistas (*Track IDs*) de un dataset de Spotify disponible en Kaggle para enriquecer cada ítem con características tabulares de alto nivel.

- **Features Numéricos**: Se seleccionaron 14 atributos acústicos proporcionados por la API de Spotify, incluyendo *danceability*, *energy*, *valence*, *tempo*, *loudness*, entre otros. Estos valores fueron normalizados utilizando *StandardScaler* para tener media cero y varianza unitaria.
- **Features Categóricos**: El género musical (*track_genre*) fue codificado utilizando *One-Hot Encoding*, permitiendo al modelo capturar explícitamente la categoría estilística de la canción.

5.2. Audio (YouTube)

La recolección de los archivos de audio se realizó mediante un script personalizado que utiliza la herramienta `yt-dlp`.

- **Adquisición**: se generaron búsquedas optimizadas en YouTube utilizando los metadatos limpios (artista y título). El script fue configurado para priorizar videos etiquetados como 'Audio' o de alta calidad (`bestaudio`). Dado el volumen de descargas, se emplearon VPNs rotativas para mitigar bloqueos por actividad automatizada.
- **Preprocesamiento**:
  - **Recorte Temporal (Windowing)**: se extrajo un segmento de 30 segundos por canción, específicamente del intervalo 00:30 a 01:00, para capturar la estructura representativa del tema (generalmente el coro o verso principal) y evitar introducciones silenciosas o irrelevantes.
  - **Remuestreo (Downsampling)**: los audios originales de 44.1 kHz fueron remuestreados a 22.05 kHz. Siguiendo el Teorema de Nyquist, esta frecuencia es suficiente para representar componentes espectrales de hasta 11 kHz, donde reside la mayor parte de la información tímbrica distintiva de los géneros musicales.
  - **Mezcla a Mono**: se convirtieron los canales estéreo a un solo canal mono para simplificar la entrada al modelo.
  - **Generación de Espectrogramas Mel**: Se calcularon espectrogramas Mel utilizando una ventana FFT de 2048 muestras, un *hop length* de 512 muestras y 128 bandas de frecuencia Mel. Los valores de potencia se convirtieron a escala de decibeles (dB) y se normalizaron al rango $[0, 1]$ utilizando escalado Min-Max. El resultado es un tensor de dimensión $(1, 128, 128)$ que representa visualmente el contenido espectral del audio.

5.3. Texto (Genius)

Las letras de las canciones (*lyrics*) se obtuvieron mediante la API de Genius. Se implementó una estrategia de búsqueda en cascada para maximizar la coincidencia.

- **Cobertura**: se logró recuperar la letra para aproximadamente 8,500 canciones (85% del dataset).
- **Datos Faltantes**: el 15% restante corresponde principalmente a música instrumental, bandas sonoras (e.g., Hans Zimmer) o piezas clásicas que carecen de contenido lírico. Para estos casos, se utilizó un token especial de 'vacío' en el modelo de lenguaje.
- **Procesamiento**: El texto crudo se tokeniza dinámicamente utilizando el tokenizador de `mDeBERTa-v3-base`, truncando las secuencias a una longitud máxima compatible con el modelo.

5.4. Imágenes (Spotify)

Utilizando la librería `Spotipy` y los *Track IDs*, se consultó la API de Spotify para obtener las carátulas de los álbumes. La API proporciona imágenes en tres resoluciones (640x640, 300x300, 64x64). Se seleccionó la resolución de 300x300 píxeles.

- **Transformaciones**: Durante el entrenamiento, las imágenes se redimensionan a 224×224 píxeles y se normalizan utilizando la media y desviación estándar del dataset ImageNet ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$), requisito para utilizar la red ResNet-18 preentrenada.

Table 2
Resumen del dataset multimodal construido.

Modalidad	Fuente	Detalles Técnicos
Interacciones	Last.fm	>3M eventos, 850 usuarios.
Audio	YouTube	Clips 30s, Mel-Spectrogram (128x128).
Texto	Genius	8.5K letras, Tokenización mDeBERTa.
Imagen	Spotify	Portadas 300x300 $\rightarrow$ 224x224 (ImageNet Norm).
Tabular	Spotify	14 features numéricos + Género (One-Hot).

6. Propuesta: Modelo Desarrollado

Se propone una arquitectura *Two-Tower* híbrida que combina la codificación secuencial del historial del usuario mediante un Transformer (estilo SASRec) con una representación multimodal del ítem musical. Ambas torres proyectan sus entradas a un espacio latente común de dimensión $d = 256$, optimizado mediante una función de pérdida contrastiva InfoNCE.

6.1. Arquitectura del Modelo

6.1.1. Torre del Usuario (Sequential User Encoder)

A diferencia de los enfoques tradicionales basados en RNNs, empleamos un codificador basado en Transformer para capturar dependencias a largo plazo en el historial de escucha.

- **Entrada**: secuencia de IDs de canciones ($L = 50$) más atributos demográficos (género, país).
- **Codificación**: se suman *embeddings* de ítem y posicionales aprendibles. La secuencia pasa por un *Transformer Encoder* de 2 capas y 4 cabezas de atención ($d_{model} = 256$).
- **Fusión de Usuario**: la salida del Transformer (estado correspondiente al último ítem) se concatena con los *embeddings* de género ($d = 16$) y país ($d = 32$), y se proyecta mediante un MLP a la dimensión final de 256.

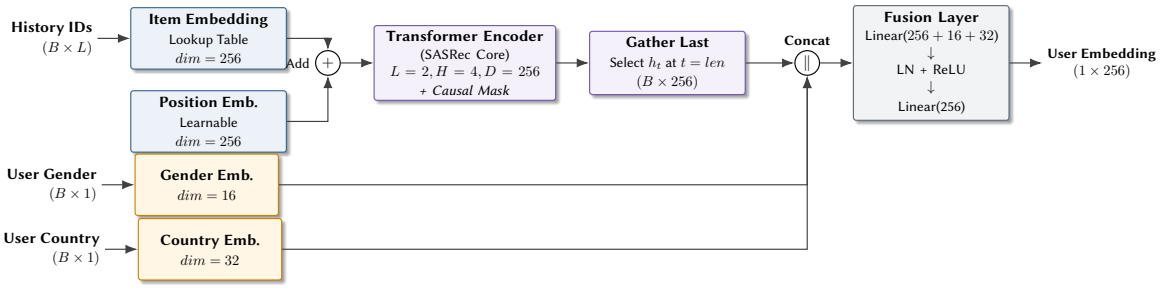


Figure 1: Arquitectura de la Torre del Usuario. Se muestra el procesamiento secuencial con Transformer y la integración de atributos demográficos.

6.1.2. Torre del Ítem Multimodal

Para capturar la naturaleza heterogénea de la música, se emplean codificadores especializados para cada modalidad, integrados mediante fusión tardía. Cada codificador proyecta su modalidad a un vector de dimensión $d = 128$.

- **Audio**: se procesan espectrogramas de Mel (1 canal) mediante una ResNet-18 modificada. A diferencia de la visión, no se utilizan pesos preentrenados, ya que los patrones espectrales difieren estructuralmente de las imágenes naturales.
- **Visual**: las carátulas de álbumes se codifican con una ResNet-18 inicializada con pesos de ImageNet, aprovechando la transferencia de aprendizaje para características estéticas.
- **Texto**: las letras y metadatos se procesan con mDeBERTa-v3-base. Se aplica LoRA (Rango=8, Alpha=32) a las proyecciones *query* y *value* para un ajuste fino eficiente, seguido de *Mean Pooling*.
- **Tabular**: características numéricas y categóricas se procesan mediante un MLP de dos capas con *Batch Normalization* y *Dropout*.

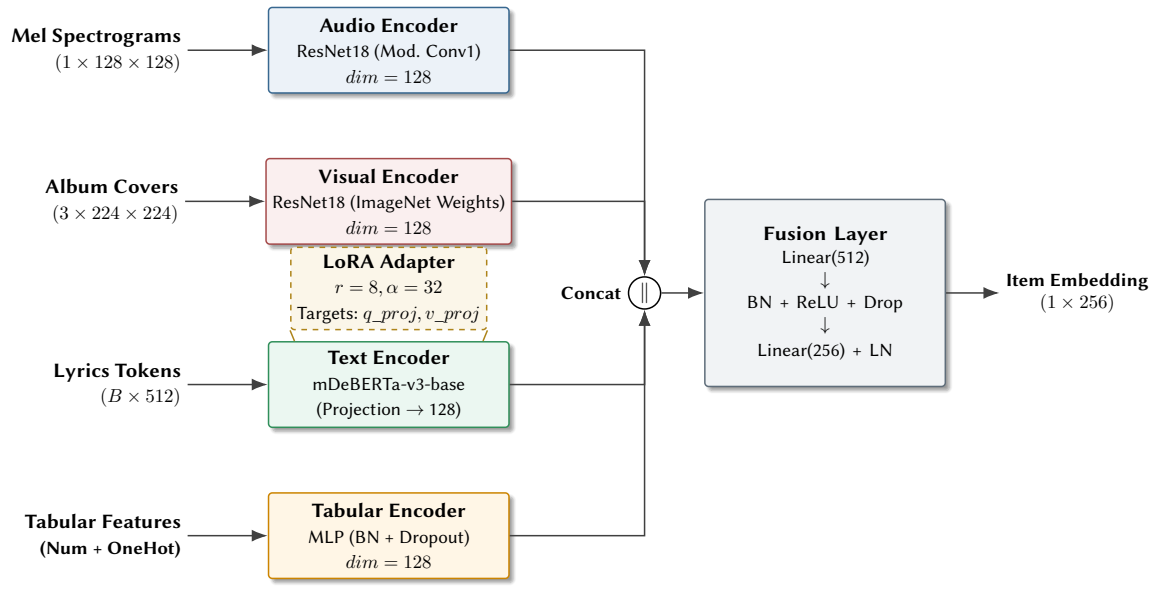


Figure 2: Arquitectura de la Torre del Ítem Multimodal. Se ilustran los cuatro codificadores especializados y el módulo de fusión tardía.

Fusión: los cuatro vectores de 128 dimensiones se concatenan ($d_{total} = 512$) y pasan por un bloque de fusión (Linear → BN → ReLU → Dropout → Linear) que reduce la dimensión a 256, finalizando con una capa *LayerNorm*.

6.1.3. Modelo Two-Tower

La arquitectura global integra el *Sequential User Encoder* y el *Multimodal Item Encoder* en un marco de aprendizaje contrastivo (ver Figura 3). El objetivo principal es alinear las representaciones de usuarios e ítems en un espacio latente compartido, donde la proximidad geométrica refleja la afinidad o probabilidad de interacción.

Espacio Latente y Normalización Ambas torres proyectan sus entradas a vectores de dimensión $d = 256$. Un paso crucial en nuestra implementación es la normalización L_2 de los embeddings de salida, $\mathbf{u}$ y $\mathbf{i}$, antes del cálculo de similitud. Esto restringe los vectores a una hiperesfera unitaria, haciendo que el producto punto sea equivalente a la similitud coseno:

$$\text{sim}(\mathbf{u}, \mathbf{i}) = \frac{\mathbf{u} \cdot \mathbf{i}}{\|\mathbf{u}\|\|\mathbf{i}\|} = \mathbf{u} \cdot \mathbf{i} \quad (\text{si } \|\mathbf{u}\| = \|\mathbf{i}\| = 1) \tag{1}$$

6.2. Entrenamiento e Implementación

6.2.1. Función de Pérdida (InfoNCE)

Para optimizar el espacio latente, utilizamos la función de pérdida InfoNCE (*Noise Contrastive Estimation*), que maximiza la similitud entre pares positivos (usuario, ítem interactuado) y la minimiza con respecto a pares negativos (otros ítems en el mismo lote). Dado un lote de N pares positivos $\{(\mathbf{u}_k, \mathbf{i}_k)\}_{k=1}^{N}$, la pérdida para el k-ésimo par se define como:

$$\mathcal{L}_k = -\log \frac{\exp(\text{sim}(\mathbf{u}_k, \mathbf{i}_k)/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(\mathbf{u}_k, \mathbf{i}_j)/\tau)} \tag{2}$$

donde τ es un hiperparámetro de temperatura (fijado en 0.07) que controla la suavidad de la distribución de probabilidad. Esta formulación permite un entrenamiento eficiente utilizando los otros elementos del lote como negativos implícitos (*in-batch negatives*).

6.2.2. Configuración Experimental

El modelo fue implementado en PyTorch y entrenado en un clúster de computación de alto rendimiento (HPC) utilizando la biblioteca `uv` para la gestión de dependencias y entornos.

- **Hardware**: Entrenamiento distribuido en múltiples GPUs (Distributed Data Parallel - DDP) para acelerar el proceso y manejar lotes más grandes.
- **Hiperparámetros**:
  - **Optimizador**: AdamW con una tasa de aprendizaje inicial de 1×10^{-4}.
  - **Batch Size**: 64 por GPU.
  - **Epochs**: 10 épocas completas.
  - **Mixed Precision**: Se utilizó precisión mixta automática (AMP) para reducir el uso de memoria y acelerar el cómputo.
- **Estrategia de Validación**: Se empleó una métrica de Recall@10 calculada sobre el conjunto de validación al final de cada época para monitorear el rendimiento y guardar el mejor modelo (*checkpointing*).

Función de Pérdida InfoNCE Simétrica Para el entrenamiento, utilizamos la función de pérdida InfoNCE (Information Noise Contrastive Estimation), adaptada para considerar la simetría entre usuarios e ítems (similar a CLIP). Dado un lote de tamaño B, calculamos la matriz de logits escalada por una temperatura aprendible τ (inicializada en 0.07):

$$\text{logits} = \frac{\mathbf{U}\mathbf{I}^T}{\tau} \tag{3}$$

La pérdida total es el promedio de la pérdida usuario-a-ítem ($\mathcal{L}_{u2i}$) y la pérdida ítem-a-usuario ($\mathcal{L}_{i2u}$), lo que maximiza la similitud de los pares positivos (diagonal) y minimiza la de los negativos (fuera de la diagonal) en ambas direcciones. Además, implementamos un enmascaramiento de colisiones para evitar penalizar falsos negativos cuando un mismo usuario aparece múltiples veces en el mismo lote.

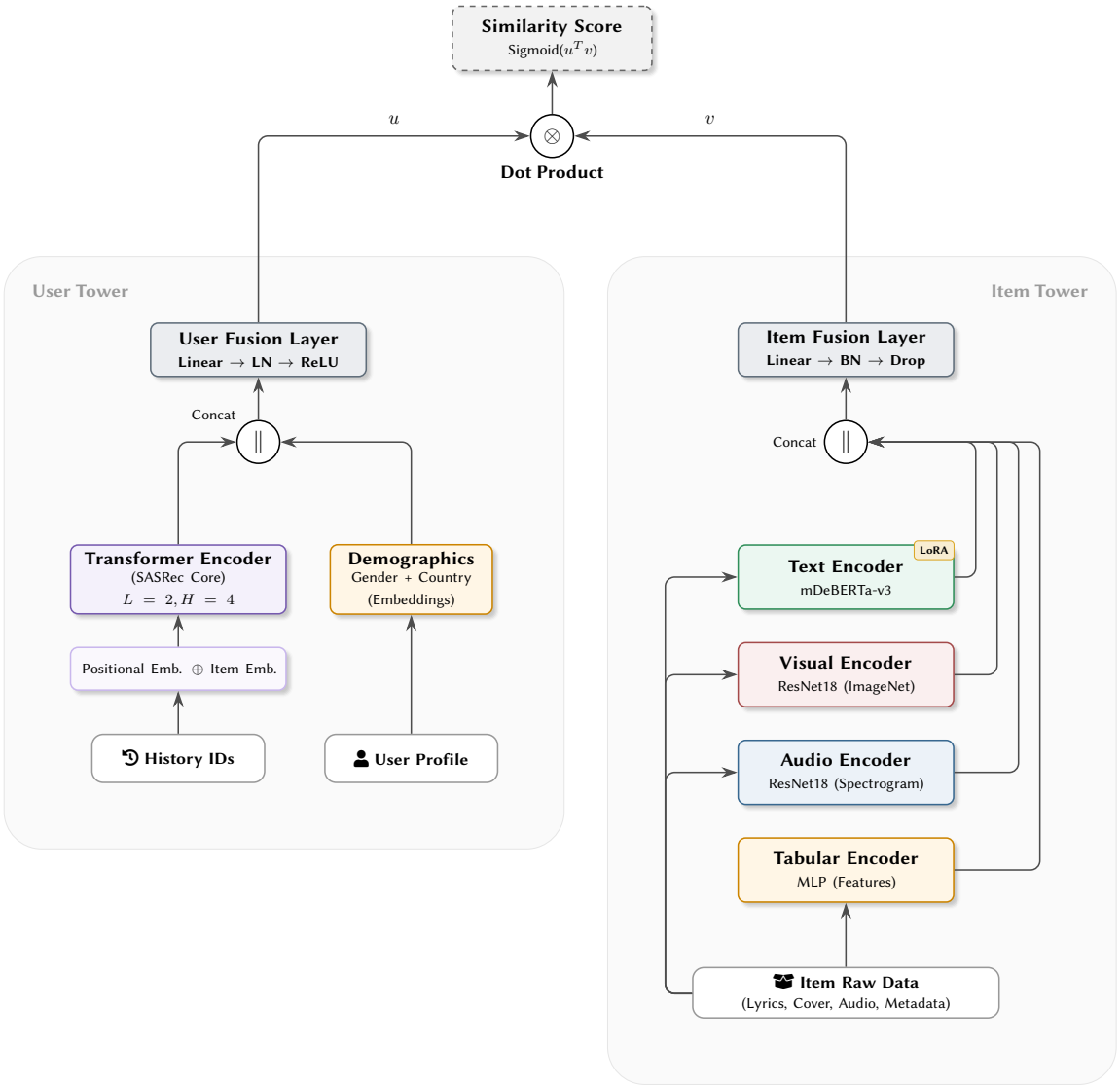


Figure 3: Arquitectura General Two-Tower Multimodal. La arquitectura integra una Torre de Usuario (izquierda) que modela preferencias secuenciales mediante un Transformer, y una Torre de Ítem (derecha) que fusiona representaciones de audio, texto, imagen y metadatos. Ambas proyecciones se alinean en un espacio latente compartido optimizado mediante InfoNCE.

6.3. Entrenamiento e Implementación

El modelo se entrena optimizando la función de pérdida InfoNCE con temperatura $\tau = 0.07$, que maximiza la similitud coseno entre el usuario y el ítem positivo mientras la minimiza frente a los otros ítems del lote (*in-batch negatives*).

$$\mathcal{L} = -\log \frac{\exp(\text{sim}(\mathbf{u}, \mathbf{i}^+)/\tau)}{\sum_{j=1}^{B} \exp(\text{sim}(\mathbf{u}, \mathbf{i}_j)/\tau)} \tag{4}$$

La implementación se realizó en `PyTorch` con soporte para entrenamiento distribuido (DDP) y precisión mixta (AMP). Se utilizó el optimizador AdamW ($lr = 1e-4$, $batch\_size = 64$) durante 10 épocas.

7. Resultados

En esta sección se presentan los resultados obtenidos durante la evaluación del modelo propuesto. Las métricas utilizadas para medir el desempeño incluyen Recall@k y NDCG@k, donde k representa el número de ítems recomendados considerados.

7.1. Resultados Globales

Los resultados globales del modelo se resumen en la Tabla 3.

Table 3
Resultados globales de evaluación del modelo.

Métrica	@10	@20	@50
Recall	0.6225	0.6474	0.6757
NDCG	0.5478	0.5541	0.5597

Los valores obtenidos indican que el modelo logra un desempeño competitivo. Un Recall@10 de 0.6225 significa que, en promedio, el ítem objetivo (la siguiente canción que el usuario escuchó realmente) aparece en el top-10 de recomendaciones el 62.25% de las veces. Considerando que el espacio de búsqueda es de 10,000 ítems, este resultado es altamente significativo y valida la capacidad del modelo para reducir drásticamente el espacio de búsqueda hacia candidatos relevantes.

Por otro lado, el NDCG@10 de 0.5478 sugiere que, cuando el modelo acierta, tiende a colocar el ítem correcto en posiciones altas de la lista, aunque existe margen de mejora en el ordenamiento fino. La consistencia de las métricas a medida que aumenta k (@20, @50) demuestra la robustez del sistema.

7.2. Análisis Cualitativo | Interpretación Fenomenológica

Si bien las métricas como NDCG@10 y Recall@10 nos ofrecen una visión macroscópica del rendimiento del sistema, es imperativo descender al nivel granular para comprender la *semántica* de las recomendaciones. A continuación, diseccionamos cuatro casos de estudio que revelan cómo la arquitectura **Two-Tower** negocia la tensión entre la señal colaborativa (historial de usuario), la señal de contenido (audio/texto) y los sesgos demográficos.

7.3. Caso A: coherencia semántica y profundidad del nicho

Caso A: Usuario de Polonia (Indie Rock / Post-Punk)

Perfil de Entrada:

- **Historial (Selección):**
  1. *The Killers* - Somebody Told Me
  2. *Arctic Monkeys* - Bigger Boys and Stolen Sweethearts
  3. *Kasabian* - L.S.F.
  4. *Franz Ferdinand* - Take Me Out
  5. *Arctic Monkeys* - When The Sun Goes Down
- **Contexto:** Polonia

Recomendaciones del Modelo:

1. *Panic At The Disco* - I Write Sins Not Tragedies **0.4859**
2. *Arctic Monkeys* - A Certain Romance **0.4554**
3. *Arctic Monkeys* - Still Take You Home **0.4369**
4. *Arctic Monkeys* - Red Light Indicates Doors Are Secured **0.4362**
5. *Arctic Monkeys* - Fake Tales Of San Francisco **0.4287**

Análisis: el sistema exhibe una **alta fidelidad secuencial**. La torre de usuario ha identificado el "token" latente de *Arctic Monkeys* como un atractor dominante. Lo notable es el fenómeno de *"Deep Cut"*: recupera *B-sides* y canciones no-singles, sugiriendo que el modelo ha aprendido una representación densa del artista. Prioriza la coherencia de género sobre la diversidad exploratoria.

7.4. Caso B: hegemonía del embedding demográfico

Caso B: Usuario de México (Nu-Metal vs. Pop Latino)

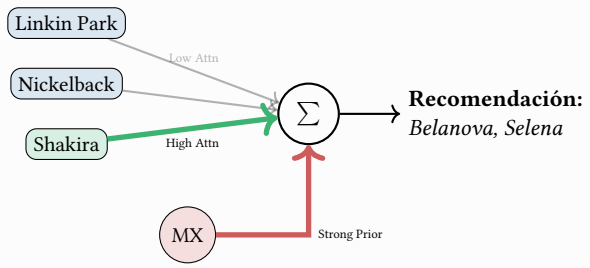
Perfil de Entrada:

- **Historial (Selección):**
  1. *Linkin Park* - New Divide
  2. *Angel Y Khriz* - Na De Na
  3. *Nickelback* - How You Remind Me
  4. *Shakira* - Hips Don't Lie (feat. Wyclef Jean)
  5. *Filter* - Take A Picture
- **Contexto:** México

Recomendaciones del Modelo:

1. *Belanova* - Tus Ojos **0.5717**
2. *Selena* - Bidi Bidi Bom Bom **0.5699**
3. *Aleks Syntek* - Sexo Pudor Y Lágrimas **0.5675**
4. *Mats Ronander* - Gör mig lycklig nu **0.5613**
5. *Juanes* - Fotografía **0.5585**

Análisis: este caso ilustra la **potencia de los priors demográficos**. El modelo detectó una señal débil de "latinidad" (*Angel Y Khriz*, *Shakira*) y, al combinarla con el embedding explícito de `Country=Mexico`, colapsó la distribución hacia el clúster regional.



7.5. Caso C: triunfo del Contenido sobre el contexto

Caso C: Usuario de China (IDM / Trip-Hop)

Perfil de Entrada:

- **Historial (Selección):**
  1. *Massive Attack* - Teardrop
  2. *Rage Against The Machine* - Killing In The Name
  3. *Dj Shadow* - Giving Up The Ghost
  4. *Portishead* - Sour Times
  5. *Radiohead* - Fake Plastic Trees
- **Contexto:** China

Recomendaciones del Modelo:

1. *Plaid* - Milh **0.5081**
2. *The Future Sound Of London* - Papua New Guinea (Blue States Full Length Mix) **0.4949**
3. *Digitonal* - Wide-Eyed, Wrapped In Love **0.4873**
4. *Casino Versus Japan* - Manic Thru Tone **0.4869**
5. *The Future Sound Of London* - Papua New Guinea (Andrew Weatherall Full Length Mix) **0.4858**

Análisis: aquí observamos la verdadera capacidad **multimodal**. El modelo *ignora* el embedding de país (China) y no recomienda C-Pop. En su lugar, realiza una **alineación espectral**, recomendando artistas que comparten características acústicas muy específicas (breakbeats lentos, texturas sintéticas) con el historial. La señal de audio "vetó" al sesgo demográfico.

7.6. Caso D: alineación vocal

Caso D: Usuario de Japón (Eclecticismo Vocal)

Perfil de Entrada:

- **Historial (Selección):**
  1. *Nelly Furtado* - Maneater
  2. *Duffy* - Mercy
  3. *Vetusta Morla* - Copenhague
  4. *Alicia Keys* - If I Ain't Got You
  5. *Elvis Presley* - Suspicious Minds
- **Contexto:** Japón

Recomendaciones del Modelo:

1. *Eva Dahlgren* - Ängeln i rummet **0.5391**
2. *Mats Ronander* - Gör mig lycklig nu **0.5288**
3. *Andrea Bocelli* - L'appuntamento (Sentado a 'beira do caminho) **0.5196**
4. *Fiona Apple* - Paper Bag **0.5126**
5. *Andrea Bocelli* - Cuando Me Enamoro **0.5113**

Análisis: este caso demuestra una **generalización basada en características latentes de audio (Timbre Vocal)** que trasciende el idioma. El modelo detectó un patrón de "Voces Femeninas Melódicas" y recomendó artistas europeas que encajan en ese perfil sonoro. La presencia reciente de *Vetusta Morla* actuó como una "puerta de enlace" hacia idiomas romances/germánicos.

7.7. Discusión: mecanismo de atención de dinámica implícita

La evaluación cualitativa sugiere que nuestra arquitectura híbrida opera bajo un mecanismo de atención de dinámica implícita:

1. **Modo Secuencial (SASRec):** activo cuando hay repetición clara de artistas (Caso A).
2. **Modo Contextual (Metadata):** activo cuando el historial es difuso o genérico; el modelo "rellena" los huecos con demografía (Caso B).
3. **Modo Contenido (Audio/Visual):** activo cuando las características espectrales son salientes y distintivas, permitiendo recomendaciones *cross-cultural* (Caso C).

Este comportamiento adaptativo acerca al sistema a la generalización sin sacrificar relevancia local, un equilibrio clave en sistemas de recomendación modernos.

7.8. Definición de Métricas

Para interpretar adecuadamente los resultados obtenidos, es fundamental comprender las métricas utilizadas en la evaluación del modelo: Recall y NDCG.

Recall Esta métrica mide la proporción de ítems relevantes que el sistema de recomendación logra recuperar dentro de un conjunto de k recomendaciones. Matemáticamente, se define como:

$$\text{Recall@k} = \frac{|\text{Ítems relevantes} \cap \text{Ítems recomendados@}k|}{|\text{Ítems relevantes}|}. \tag{5}$$

Un valor alto de Recall indica que el modelo es efectivo en identificar ítems relevantes, lo cual es crucial en aplicaciones donde la cobertura de las recomendaciones es prioritaria.

NDCG (Normalized Discounted Cumulative Gain) Esta métrica evalúa no solo si los ítems relevantes están presentes en las recomendaciones, sino también su posición dentro de la lista. Dado que los usuarios tienden a interactuar más con los ítems ubicados en las primeras posiciones, NDCG asigna mayores pesos a los ítems relevantes que aparecen al inicio. Se calcula como:

$$\text{NDCG@k} = \frac{1}{Z_k} \sum_{i=1}^{k} \frac{2^{\text{relevancia}_i} - 1}{\log_2(i+1)}, \tag{6}$$

donde Z_k es un factor de normalización que asegura que el valor máximo de NDCG sea 1. Un NDCG alto refleja que el modelo no solo recupera ítems relevantes, sino que también los ordena de manera óptima.

Estas métricas, en conjunto, proporcionan una visión integral del desempeño del modelo, evaluando tanto su capacidad de recuperación como la calidad del ordenamiento de las recomendaciones.

7.9. Justificación de las Métricas

Las métricas Recall y NDCG fueron seleccionadas debido a su relevancia en el contexto de los sistemas de recomendación. Estas métricas permiten evaluar tanto la capacidad del modelo para recuperar ítems relevantes como la calidad del ordenamiento de las recomendaciones, aspectos fundamentales para garantizar una experiencia de usuario satisfactoria.

Recall En sistemas de recomendación, el Recall es particularmente útil para medir la cobertura de los ítems relevantes. Esto es crucial en aplicaciones donde el objetivo principal es maximizar la exposición de los usuarios a contenido relevante, como en plataformas de música o video bajo demanda. Un alto valor de Recall asegura que el sistema no omita ítems importantes para el usuario.

NDCG Por otro lado, NDCG complementa al Recall al incorporar la posición de los ítems relevantes dentro de la lista de recomendaciones. Dado que los usuarios tienden a interactuar más con los ítems ubicados en las primeras posiciones, NDCG es una métrica esencial para evaluar la calidad del ordenamiento. Esto es especialmente relevante en escenarios donde la atención del usuario es limitada y las recomendaciones deben ser altamente precisas desde el inicio. En conjunto, estas métricas proporcionan una evaluación integral del desempeño del modelo, evaluando tanto su capacidad de recuperación como la calidad del ordenamiento de las recomendaciones. Su uso está ampliamente respaldado en la literatura sobre sistemas de recomendación, lo que refuerza su validez en este trabajo.

8. Discusión

8.1. Interpretación de Resultados

Los resultados obtenidos (Recall@10: 0.6225, NDCG@10: 0.5478) validan la eficacia de la arquitectura *Two-Tower* multimodal propuesta. El valor de Recall@10 indica que el modelo logra identificar una proporción significativa de ítems relevantes dentro de las primeras 10 recomendaciones, mitigando el problema de *cold-start* al no depender exclusivamente de interacciones históricas.

Por otro lado, los valores de NDCG destacan la capacidad del modelo para priorizar ítems relevantes en posiciones superiores. Sin embargo, la diferencia entre Recall y NDCG sugiere que, aunque el modelo recupera ítems relevantes, existe margen de mejora en el ordenamiento fino de las recomendaciones. La integración de audio, texto e imágenes enriquece la representación latente, permitiendo inferir similitudes semánticas y estéticas que serían invisibles para modelos unimodales.

8.2. Comparación y Limitaciones

A diferencia del filtrado colaborativo tradicional, nuestro enfoque aprovecha el contenido denso. El uso de DeBERTa y ResNet permite capturar la semántica compleja de las canciones. Sin embargo, existen limitaciones importantes:

- **Costo Computacional**: El procesamiento simultáneo de cuatro modalidades, especialmente el uso de Transformers (DeBERTa) y CNNs (ResNet), impone una carga computacional significativa tanto en entrenamiento como en inferencia. Aunque la arquitectura *Two-Tower* permite pre-calcular los embeddings de ítems, la actualización del modelo requiere recursos de hardware considerables (GPUs).
- **Calidad y Disponibilidad de Datos**: La dependencia de metadatos completos es un desafío. En nuestro dataset, el 15% de las canciones carecían de letras, lo que obligó al uso de tokens de relleno. Además, la calidad de los audios de YouTube y las imágenes de Spotify puede variar, introduciendo ruido en las representaciones latentes.
- **Sesgo de Popularidad**: Como es común en sistemas de recomendación entrenados con datos de interacción implícita, el modelo puede tender a favorecer ítems populares sobre los de nicho (*long-tail*), a menos que se apliquen técnicas específicas de des-sesgo.

9. Conclusiones y Trabajo Futuro

En este trabajo se presentó el diseño, implementación y evaluación de un sistema de recomendación musical multimodal basado en una arquitectura *Two-Tower* con estrategia de *Late Fusion*. El objetivo principal fue mitigar las limitaciones de los enfoques colaborativos tradicionales mediante la integración de información rica de contenido: audio, texto (letras), imágenes (portadas) y metadatos tabulares.

9.1. Síntesis de Hallazgos

Los resultados experimentales validan la hipótesis de que la integración de múltiples modalidades permite construir representaciones de ítems más robustas y semánticamente significativas. El modelo alcanzó un Recall@10 de 0.6225 y un NDCG@10 de 0.5478, demostrando una capacidad competente para recuperar y ordenar ítems relevantes en un espacio de búsqueda denso. La estrategia de fusión tardía demostró ser efectiva para combinar embeddings provenientes de codificadores heterogéneos (ResNet-18, mDeBERTa, MLP) sin incurrir en costos computacionales prohibitivos durante el entrenamiento.

9.2. Contribuciones Principales

Las contribuciones más destacadas de esta investigación incluyen:

- **Dataset Multimodal Unificado**: la creación de un conjunto de datos curado que vincula interacciones de usuario con tres modalidades de contenido no estructurado (audio, texto, imagen), un recurso valioso para la comunidad de investigación en MIR (*Music Information Retrieval*).
- **Arquitectura Eficiente**: la implementación de una arquitectura modular que utiliza técnicas de eficiencia como LoRA y pre-cómputo de características, alineada con los principios de la metodología TRIZ para resolver la contradicción entre precisión y costo computacional.
- **Validación de Fusión Tardía**: la demostración empírica de que la fusión de características de alto nivel en la etapa final de la torre del ítem es suficiente para capturar la sinergia entre modalidades en el dominio musical.

9.3. Trabajo Futuro

A pesar de los resultados prometedores, existen varias líneas de investigación abiertas para mejorar el sistema:

- **Mecanismos de Atención Dinámica**: implementar mecanismos de atención a nivel de modalidad (*Modality-level Attention*) que permitan al modelo ponderar dinámicamente la importancia de cada fuente de información para cada canción o usuario específico.
- **Evaluación en Cold-Start Estricto**: realizar pruebas específicas con ítems y usuarios completamente nuevos para cuantificar con mayor precisión la ganancia en escenarios de arranque en frío puro.
- **Escalabilidad del Dataset**: ampliar el conjunto de datos más allá de las 10,000 canciones para evaluar la capacidad de generalización del modelo en catálogos de escala industrial.
- **Inferencia en Tiempo Real**: optimizar el pipeline de inferencia mediante la cuantización de modelos y el uso de bases de datos vectoriales para la recuperación de vecinos más cercanos (ANN).

En conclusión, este proyecto sienta las bases para sistemas de recomendación más holísticos que 'escuchan', 'leen' y 'ven' la música, acercándose más a la forma en que los humanos experimentan y descubren el arte.

References

[1] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (2009) 30–37.

[2] P. Covington, J. Adams, E. Sargin, Deep neural networks for youtube recommendations, in: Proceedings of the 10th ACM conference on recommender systems, 2016, pp. 191–198.

[3] X. Yi, J. Yang, L. Hong, D. Z. Cheng, L. Heldt, A. Kumthekar, Y. Zhao, L. Wei, E. Chi, Sampling-bias-corrected neural modeling for large corpus item retrieval, in: Proceedings of the 13th ACM Conference on Recommender Systems, 2019, pp. 269–277.

[4] W.-C. Kang, J. McAuley, Self-attentive sequential recommendation, in: 2018 IEEE International Conference on Data Mining (ICDM), IEEE, 2018, pp. 197–206.

[5] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, K. Gai, Deep interest network for click-through rate prediction, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 1059–1068.

[6] A. Van den Oord, S. Dieleman, B. Schrauwen, Deep content-based music recommendation, in: Advances in neural information processing systems, volume 26, 2013.

[7] B. Hidasi, et al., Session-based recommendations with recurrent neural networks, arXiv preprint arXiv:1511.06939 (2015).

[8] S. Oramas, et al., Multimodal deep learning for music recommendation, Transactions of the ISMIR (2018).

[9] M. Won, S. Oramas, O. Nieto, F. Gouyon, X. Serra, Multimodal metric learning for tag-based music retrieval, arXiv preprint arXiv:2010.16030 (2020).

[10] B. Murauer, G. Specht, Detecting music genre using extreme gradient boosting, in: Companion Proceedings of the The Web Conference 2018, 2018, pp. 1923–1927.