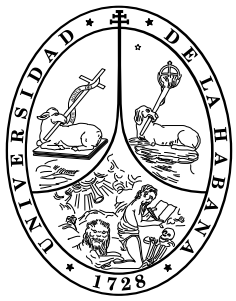
Facultad de Matemática y Computación

Universidad de La Habana



Tesis de diploma de la carrera Ciencia de la Computación

Recuperación semántica de música utilizando embeddings y modelos de clasificación.

Autor: **Niley González Ferrales**

Tutor: **Dr. Yudivian Almeida**

La Habana

2024

A la ciencia.

A los gigantes en cuyos hombros nos alzamos para elevar a la humanidad a nuevas alturas.

Opinión del tutor

Dr. Yudivian Almeida
Facultad de Matemática y Computación
Universidad de la Habana
Enero, 2023

Índice general

Resumen

El lenguaje natural, como una de las interfaces más intuitivas conocidas por los humanos, tiene un potencial significativo como intermediario de muchas tareas que involucran la interacción humano-computadora, especialmente en campos enfocados en aplicaciones como la Recuperación de Información Musical (MIR, por sus siglas en inglés). El objetivo de este estudio fue investigar y diseñar un prototipo para la recuperación de música utilizando texto. Esto implicó recuperar contenido musical de un conjunto de candidatos que mejor coincidiera con una descripción dada en lenguaje natural, una tarea que ha recibido atención limitada en la literatura existente.

Este trabajo exploró un enfoque basado en modelos de MIR para extraer características de bajo y alto nivel de un archivo de música. Estas características se utilizaron para generar descripciones en lenguaje natural. Para recuperar, se utilizaron *embeddings* de BERT (*Bidirectional Encoder Representations from Transformers*) para calcular la similitud semántica entre la consulta del usuario y las descripciones de la base de datos. El resultado produjo un prototipo prometedor, estableciendo una base sólida para futuros esfuerzos en esta línea de investigación.

Palabras claves: recuperación de información musical, recuperación basada en texto, descripción de música, vectores de *embeddings*, Sentence BERT

Abstract

Natural language, as one of the most intuitive interfaces known to humans, holds significant potential in mediating many tasks involving human-computer interaction, particularly in application-focused fields like Music Information Retrieval (MIR). The aim of this study was to research and design a prototype for text-to-music retrieval. This involved retrieving music content from a pool of candidates that best matched a given natural language description, a task that has received limited attention in existing literature.
This work explored an approach that relied on MIR models to extract low-level and high-level features from a music file. These features were employed to generate natural language descriptions. For retrieval purposes, BERT embeddings were used to calculate semantic similarity between the user query and the database descriptions. The outcome yielded a promising prototype, laying a strong foundation for future efforts in this line of research.

Keywords: music information retrieval, text-based retrieval, music captioning, embeddings vectors, Sentence BERT

Capítulo 1

Introducción

El auge de la tecnología digital, en los últimos años, ha llevado a una explosión en la cantidad de contenido multimedia accesible en línea. Incluyendo texto (libros, páginas web, ...), imágenes, videos y música; la red se ha convertido en un repositorio de información con un crecimiento vertiginoso [1]. La necesidad de buscar en bases de datos cada vez más grandes ha impulsado importantes avances en el campo de la búsqueda y recuperación de información. Existen motores de búsqueda para diversos contenidos como páginas webs, imágenes y videos; sin embargo la música no es accesible de la misma forma [2]. En este contexto, la presente tesis se propone explorar nuevas técnicas y enfoques para la búsqueda y recuperación de música en grandes bases de datos, con el objetivo de mejorar la accesibilidad de este recurso.

1.1. Contexto histórico/social

Durante las últimas décadas la tecnología computacional se ha desarrollado al punto de estar presente, de una manera u otra, en casi todos los procesos cotidianos de los seres humanos. La velocidad a la que se generan grandes volúmenes de datos (conocido comunmente como *big data*), en la actualidad, desafía las capacidades de procesamiento computacional [3].
Una característica fundamental de estos volúmenes de datos es la gran variedad que presentan, siendo alrededor del 80% de naturaleza no estructurada. Y precisamente, la recuperación de información se define como encontrar resultados de naturaleza no estructurada que satisfaga una necesidad de información dentro de una gran colección de datos [4].

Los sistemas de recuperación de información (SRI) consisten en tecnologías y métodos diseñados para la búsqueda, almacenamiento, recuperación y organización de información. Estos sistemas son esenciales en la gestión de grandes cantidades de datos en diversos entornos, como bibliotecas digitales, bases de datos en línea y motores de búsqueda en la web.
La recuperación de información utilizando computadoras se remonta a la década de 1950, y desde entonces se han desarrollado grandes ideas en el campo. Entre estas ideas se encuentran la creación de rankings de documentos, la representación vectorial de documentos y consultas, el agrupamiento de documentos similares, la asociación de términos con similitudes semánticas, la introducción de la frecuencia inversa en documentos y los modelos de semántica latente [5].
Uno de los avances recientes en la recuperación de información es el uso de redes neuronales. Este es un campo que ha avanzado a pasos agigantados en la última década, en gran parte gracias al aumento del poder computacional introducido con las GPU. Los modelos de lenguaje, como Word2Vec [6], GloVe [7] y los más recientes, basados en *transformers* [1], como BERT [8] y GPT [9], han demostrado una capacidad excepcional para capturar el significado semántico y la relación contextual entre palabras y frases. Estos modelos operan con representaciones de *embeddings* a partir de diferentes enfoques cada uno.
Los embeddings son representaciones, en un espacio de relativamente baja dimensionalidad, de *tokens* como frases, párrafos o documentos, a partir de un espacio vectorial de alta dimensionalidad, donde cada dimensión corresponde a una característica o atributo del lenguaje aprendido. Al aplicar estos *embeddings* a la representación de documentos y consultas, se ha logrado una mejora significativa en la precisión y la relevancia de los resultados de recuperación de información [10].

1.2. Motivación

A pesar de que los SRI se han centrado principalmente en la recuperación de información con forma textual, se ha reconocido la necesidad de adaptar el entorno para manejar todo tipo de datos (como imágenes, vi-

[1] los *transformers* son arquitecturas de aprendizaje profundo basadas en el mecanismo de autoatención. Se emplean, en especial, en el campo de procesamiento automático del lenguaje y visión artificial. Sus versiones más recientes se han utilizado predominantemente para entrenar grandes modelos de lenguaje en grandes datasets

deos, audios) [11]. La recuperación de información en datos no textuales presenta desafíos únicos, como la necesidad de comprender el contenido visual o auditivo de los datos. En el caso de imágenes, por ejemplo, la búsqueda puede basarse en características visuales como el color, la textura o la forma. En el caso de videos y audios, la búsqueda puede basarse en características como el tono, el ritmo o el contenido verbal. En los últimos años, ha habido avances significativos en la recuperación de información en datos no textuales, impulsados por el desarrollo de algoritmos de *deep learning* y el aumento de la capacidad computacional. Estos avances han permitido el desarrollo de sistemas más precisos y eficientes para la recuperación de imágenes, videos y audios.

Una herramienta utilizada comúnmente para recuperar información multimedia es representar los datos como texto, empleando metadatos [12]. Estos metadatos sirven para describir y normalizar la representación de la información digital, lo que a su vez facilita su posterior búsqueda y recuperación. A pesar de sus beneficios, el uso de metadatos fijos conlleva limitaciones significativas como la imposibilidad de adaptarse a cambios en la información y la falta de precisión en la descripción de la información, así como la restricción a una cantidad limitada de información descriptiva. En este sentido, los metadatos fijos pueden resultar insuficientes para abarcar la complejidad inherente a la información multimedia, lo que a su vez puede dificultar su precisa recuperación.

La creación de metadatos suele requerir un enfoque manual o, en su defecto, la utilización de modelos de inteligencia artificial. Sin embargo, estos modelos, al depender de los datos con los que fueron entrenados, sufren dificultades para generalizar adecuadamente para contextos específicos. Además, la naturaleza dinámica de la información multimedia puede llevar a que los metadatos fijos se vuelvan obsoletos con el tiempo, lo que impacta negativamente la precisión y exhaustividad en la recuperación de información.

Dadas las limitaciones que presenta la recuperación utilizando metadatos, se han estudiado otras alternativas. Un ejemplo es la consulta por tarareo (*query-by-humming*) o la consulta por ejemplo (*query-by-example*), en la que la consulta se representa mediante una grabación de audio y el sistema recupera la canción correcta (la más similar). La recuperación basada en contenido consiste en buscar audio que coincida con una consulta de audio. Dado un audio de ejemplo, se devuelven los audios más similares en la base de datos. Sin embargo, satisfacer los requisitos para introducir un audio puede resultar difícil para los usuarios. Por otro lado, las consultas detalladas en lenguaje natural con forma libre constituyen una interfaz fa-

miliar ampliamente utilizada en los motores de búsqueda actuales, lo que permite solicitar música con información semántica.
Permitir a los usuarios buscar música utilizando oraciones en lenguaje natural introduce una capa adicional de complejidad, ya que requiere el puente entre la semántica del lenguaje y la representación de la música en sí misma.
Uno de los desafíos clave en la integración del lenguaje natural y análisis de audios en los SRI se encuentra en la representación efectiva de la música en función de las consultas textuales. Hasta donde se ha podido comprobar, este problema no ha sido abordado utilizando las emergentes capacidades semánticas y contextuales de los grandes modelos de lenguajes (LLM por sus siglas en inglés). Tampoco se ha observado una estrategia de caracterización de la música empleando modelos pre-entrenados de inteligencia artificial, como una alternativa a entrenar modelos de *music captioning* que son la forma más directa de representar música como texto. Depender de un único modelo trae desventajas en cuanto a que las características que se pueden extraer, están limitadas a las que aparecen en los datos de entrenamiento; esas desventajas son acentuadas por el conocido problema que hay con la indisponibilidad de *datasets* de tamaño adecuado y con amplio rango de modalidades [13].

1.3. Antecedentes

Este trabajo es pionero en la investigación de recuperación semántica de música en la Facultad. No se cuentan con investigaciones en esta área.

1.4. Problemática

Durante una exahustiva investigación no se encontró ningún sistema o diseño que permita organizar y recuperar música de una gran base de datos utilizando consultas en lenguaje natural y con componentes semánticos y que, además, sea escalable respecto a las características(o *features*) que se conocen de la música.

1.5. Objetivos

El objetivo de esta investigación es el diseño e implementación de un prototipo de una plataforma que permita realizar consultas en lenguaje natural sobre una base de datos de música y obtener resultados con significación semántica.
Los objetivos específicos planteados para dar cumplimiento al objetivo general son:

- Estudiar los resultados más recientes relacionados con recuperación de información multimedia, en particular de música. Indagando en recuperación con consultas en lenguaje natural.
- Modelar una sistema de extracción de *features* que sea flexible, permitiendo actualizar y añadir información sobre la música.
- Modelar un SRI que devuelva canciones relevantes a consultas en lenguaje natural. Intentando incluir similitud semántica sin sacrificar mucha eficiencia.
- Implementar la plataforma modelada anteriormente.
- Evaluar el Sistema de Recuperación y comparar los resultados con modelaciones similares en la literatura.
- Realizar una página web utilizando django *framework* que permita escribir consultas y reproducir la música que sea resultado de la consulta; como interfaz para probar el prototipo implementado.

1.6. Estructura de la tesis

A continuación se describe la estructura del documento. En el capítulo 2, **Estado del Arte**, se define el marco teórico del trabajo, que abarcará los modelos de recuperación de información, los acercamientos a recuperación de audios y música, modelos de lenguaje y como sobreponerse al *semantic gap*. En el capítulo 3, **Diseño e implementación**, se presenta la modelación de la propuesta para atacar el problema planteado y se describen los detalles de implementación incluyendo las alternativas que no se pudieron comprobar. Finalmente, en el capítulo 4, **Breve Experimentación**, se muestran las evaluaciones del modelo de recuperación. A continuación aparecen las **Conclusiones** y las **Recomendaciones**. El documento finaliza con la **Bibliografía**.

Capítulo 2

Estado del Arte

Este trabajo se relaciona con varios temas en la literatura: procesamiento y recuperación de información musical, recuperación de música basada en texto, brecha semántica, procesamiento de lenguaje natural en Recuperación de Información Musical (MIR, por sus siglas en inglés), descripción de música (*music captioning*), aprendizaje a partir de supervisión de lenguaje y *embeddings* en recuperación de información.

2.1. Procesamiento y recuperación de información musical

El campo de investigación de MIR se ocupa de la extracción e inferencia de características significativas de la música (ya sea desde las señales de audio, representación simbólica o fuentes externas como páginas web), indexando la música usando estas características y desarrollando diferentes esquemas de búsqueda y recuperación (por ejemplo, búsqueda basada en contenido, sistemas de recomendación musical o interfaces de usuario para explorar grandes colecciones de música) [14].

La música es un artefacto humano altamente multimodal. En este caso, por modalidad, se refiere a una forma específica de digitalizar la información musical [13]. Diferentes modalidades se obtienen a través de diferentes transductores, en diferentes lugares o momentos, y/o pertenecen a diferentes medios. Ejemplos de modalidades que pueden asociarse a una sola pieza de música incluyen audio, letras, partituras simbólicas, portadas de álbumes, y así sucesivamente [13, 14].

Los enfoques computacionales en MIR típicamente emplean características y crean modelos para describir la música por una o más de las siguientes categorías de percepción musical: contenido musical, contexto musical, propiedades del usuario y contexto del usuario, como se define en *Music Information Retrieval: Recent Developments and Applications* [14]. Se hace referencia al contenido musical como los aspectos codificados en la señal de audio, tales como: ritmo, melodía, volumen, letras de canciones y timbre. Mientras que el contexto musical se define como factores que no pueden extraerse directamente del audio, pero que están relacionados, no obstante, con el ítem musical, el artista o el intérprete. Al centrarse en el usuario, los aspectos del contexto del usuario representan factores dinámicos y frecuentemente cambiantes, y las propiedades del usuario se refieren a características constantes o de cambio lento, como sus gustos musicales o educación musical.

Es importante señalar que existen interconexiones entre algunas características de diferentes categorías. Por ejemplo, los aspectos reflejados en el contexto musical (por ejemplo, el género musical) pueden ser modelados por el contenido musical (por ejemplo, la instrumentación).

Aplicaciones de procesamiento y recuperación de información musical

- La recuperación de música tiene la intención de ayudar a los usuarios a encontrar música en grandes colecciones según un criterio particular de similitud. En *query-by humming* (búsqueda por tarareo) y *query-by example* (búsqueda mediante ejemplo), el objetivo es recuperar música a partir de una entrada melódica o rítmica dada. Se basan en la comparación de una señal musical objetivo con una base de datos, pero los usuarios pueden querer encontrar música que cumpla con ciertos requisitos (por ejemplo, "dame canciones con un tempo de 100 bpm o en do mayor") [14]. De hecho, las personas generalmente usan etiquetas o descripciones semánticos (como ‘alegre’ o ‘rock’) para referirse a la música.

- El alineamiento o sincronización de audio es un escenario similar a la recuperación de música donde, además de identificar un fragmento de audio dado, el objetivo es conectar localmente, posiciones temporales de dos señales musicales.

- Los sistemas de recomendación de música suelen proponer una lista de piezas musicales basándose en modelar las preferencias musicales del usuario.

- La generación automática de listas de reproducción está relacionada con la recomendación de música. Su objetivo es crear una lista ordenada de resultados, como pistas musicales o artistas, para proporcionar listas de reproducción significativas y agradables para el oyente. Una de las diferencias entre la recomendación de música y la generación de listas de reproducción es que la primera suele proponer nuevas canciones no conocidas por el usuario, mientras que la segunda tiene como objetivo reorganizar material ya conocido.

Tareas de procesamiento y recuperación de información musical

- Extracción de características del contenido y contexto musical. Las características de audio pueden subdividirse en físicas y perceptuales [15]. Las físicas, que pueden calcularse en diversos dominios como el tiempo, la frecuencia o la wavelet, incluyen *zero-crossing rate*, la amplitud, el ritmo, basadas en autoregresión, basadas en STFT (Transformada de Fourier de Tiempo Corto), brillo, tonalidad, croma y forma del espectro. Por otro lado, las características perceptuales intentan integrar el procesamiento de percepción del sonido humano. Por ejemplo, los Coeficientes Cepstrales de Frecuencia Mel (MFCC), paquetes de wavelets perceptuales, y la intensidad sonora.

- La similitud es la tarea de calcular la la semejanza entre el contenido de la información. A menudo, esta tarea tiene el propósito de recuperar documentos de una colección a través de una consulta, que puede ser explícitamente expresada por el usuario o deducida implícitamente por el sistema [13]. Un ejemplo muy común de consultas de similitud explícita es la búsqueda por ejemplo, en la cual la consulta está representada por una grabación de audio y el sistema recupera la canción correcta. Por otro lado, las consultas implícitas se utilizan en sistemas de recomendación y generadores de listas de reproducción.

- La clasificación consiste en tomar como entrada una pieza musical y devolver una o más etiquetas. Una tarea de clasificación popular es el reconocimiento del estado de ánimo o la emoción [16], mientras que una emergente es la clasificación de género [17, 18, 19, 20, 21, 22, 23]. Ambas tareas pueden aprovechar grabaciones de audio, letras, portadas y metadatos. Otras tareas de clasificación incluyen: clasificación de instrumentos, clasificación de obras derivadas, identificación de tonos y descripción musical expresiva.

2.2. Recuperación de música basada en texto

Con el paso de los años se han propuesto numerosos enfoques para navegar, buscar y descubrir música a través de una variedad de interfaces [24]. Más allá de la búsqueda simple por metadatos, los sistemas de recuperación de música permiten expresar consultas a través de letras [25], ejemplos de audio [26], videos [27] y tarareos [28], entre otros.

A pesar de que cada uno de estos tipos de consulta tiene sus méritos, ninguno de ellos admite una de las formas más populares de buscar en la actualidad: mediante texto libre. Por ejemplo, comúnmente buscamos canciones escribiendo texto en un motor de búsqueda o preguntando a comunidades en línea para identificar una pieza de música de la que no tenemos información bibliográfica o editorial. Habilitar a los sistemas MIR para interpretar consultas en lenguaje natural puede tener beneficios de gran alcance [24].

Un sistema de recuperación basado en texto ideal necesita ser flexible para permitir varios tipos de entrada (como palabras, oraciones) y tener vocabularios abundantes. Por ejemplo, se pueden utilizar etiquetas populares como el género, para explorar la biblioteca musical. A veces, las consultas de entrada pueden incluir tipos de etiquetas musicales no vistas anteriormente. Además, uno puede utilizar descripciones más detalladas a nivel de oración para descubrir música.

La recuperación basada en texto es desafiante porque necesita manejar, no solo metadatos editoriales (título, artista, año de lanzamiento), sino también información semántica (género, estado de ánimo, tema). Además, los sistemas modernos de recuperación, como los asistentes de voz, necesitan generalizar a entradas de lenguaje natural a nivel de oración; más allá de vocabularios con etiquetas fijas [12].

Otro enfoque reciente para la recuperación de información (IR) es a través de modelos neuronales de ranking, que utilizan redes neuronales superficiales o profundas. Los modelos tradicionales de aprendizaje de clasificación emplean técnicas de aprendizaje automático sobre *features* hechos a mano. En contraste, los modelos neuronales aprenden representaciones del lenguaje a partir de texto puro; que pueden disminuir la brecha entre el vocabulario de la consulta y el del documento [29].

Desde el comienzo de la década, ha habido mejoras dramáticas en el rendimiento en tareas de visión por computadora, reconocimiento de voz y traducción automática, tanto en la investigación como en aplicaciones del mundo real [30]. Estos avances han sido propulsados en gran medida por los recientes progresos en modelos de redes neuronales, generalmente

con múltiples capas ocultas, conocidos como arquitecturas profundas (*deep learning*) [30, 31, 32, 33].

Los modelos neuronales para IR utilizan representaciones vectoriales de texto y suelen contener una gran cantidad de parámetros que necesitan ser ajustados. Los modelos de aprendizaje automático (machine learning, ML) con un gran conjunto de parámetros suelen requerir una gran cantidad de datos de entrenamiento [34].

El aprendizaje de representaciones adecuadas de texto, también requiere datasets a gran escala durante el entrenamiento [35]. Por lo tanto, a diferencia de los modelos clásicos de IR, estos enfoques neuronales tienden a requerir muchos datos; mejorando su rendimiento con la cantidad de datos de entrenamiento [29].

Los enfoques más similares a esta investigación son los estudios en MusCALL [24] y MuLan [36], que utilizan aprendizaje contrastivo cruzado y multimodal para crear un espacio compartido de *embeddings* o *embeddings* multimodales. Publicado en noviembre de 2022, *Toward Universal Text-To-Music Retrieval* [12] ahonda en un estudio de diseños efectivos para sistemas de recuperación de texto-música, proponiendo un sistema universal de recuperación de texto-música, que logra un rendimiento de recuperación comparable tanto en consultas a nivel de *tags* como a nivel de oraciones.

2.3. Brecha semántica

La brecha semántica (*semantic gap*) en la recuperación de música se refiere a la diferencia entre descripciones acústicas de bajo nivel y conceptos humanos de alto nivel (significativos para la percepción musical humana). Por ejemplo, el mismo tempo puede aparecer en diferentes géneros musicales, y un género musical dado puede estar caracterizado por diferentes tempos [37].

Esta brecha existe porque las características que los ordenadores pueden analizar, como el tono, el tempo y el volumen, no son los mismos que los conceptos que los humanos utilizan para relacionarse con colecciones de música [38]. Idealmente, los enfoques de recuperación y recomendación de música deberían incorporar aspectos de varias categorías para superar la brecha semántica [14].

Un enfoque multimodal para superar la brecha semántica en la música implica combinar diversas técnicas y fuentes de datos para mejorar la precisión de los sistemas de recuperación y recomendación de música [39].

2.4. Procesamiento de Lenguaje Natural en MIR

El Procesamiento de Lenguaje Natural (NLP, por sus siglas en inglés) es un campo de la Ciencia de la Computación y la Inteligencia Artificial, que se ocupa de la interacción entre las computadoras y el lenguaje humano (natural) [40]. El NLP es un componente fundamental en tecnologías de la vida diaria: búsqueda en la web, reconocimiento y síntesis de voz, resúmenes automáticos en la web, recomendación de productos (incluida la música), traducción automática, entre otros.

Trabajos previos en la literatura de MIR han explorado aprovechar NLP en el ámbito musical desde diferentes enfoques. Los esfuerzos iniciales se centraron en el texto como una modalidad aislada, adoptando técnicas para construir bases de conocimiento a partir de corpus de texto relacionados con la música [41], o construir grafos semánticos para la similitud entre artistas a partir de biografías [42]. Los esfuerzos recientes han comenzado a favorecer, en cambio, enfoques multimodales. Estos han explorado el aprendizaje profundo con datos de entrada multimodales, típicamente audio combinado con texto, como reseñas o letras, para aplicaciones tan variadas como la clasificación y recomendación musical [43], detección de estado de ánimo [44], reconocimiento de emociones en la música [45] y descripción de música (captioning) [46, 47].

Por otro lado, la idea de permitir que los sistemas de MIR interpreten consultas en lenguaje natural no es nueva; algunos trabajos, como el de Brian Whitman y Ryan M. Rifkin [48], han sugerido direcciones de investigación similares. Sin embargo, hasta recientemente, los sistemas multimodales que integran lenguaje natural no han sido ampliamente adoptados dentro de la comunidad de MIR, posiblemente debido a la falta de datasets adecuados o a las limitaciones prácticas de los métodos de NLP anteriores a los modelos de lenguaje modernos [24]. A la luz de los avances recientes en modelos de lenguaje, se argumenta que el aprendizaje audio-lenguaje tiene el potencial de cerrar la brecha semántica en MIR.

2.5. Descripción de música

Music captioning se define como la tarea de generar una descripción en lenguaje natural del contenido de audio de la música de una manera humana. MusCaps [46] en 2021, afirman presentar el primer modelo de *music audio captioning*. Hasta hace poco, los enfoques en MIR para la descripción de música normalmente se basaban en clasificación de una o múltiples eti-

quetas. Un ejemplo destacado es el *auto-tagging* de música [49, 50, 51], en el cual se asignan *tags* descriptivos a un fragmento musical para transmitir características de alto nivel de la entrada, como género, instrumentación y emoción.

Los sistemas de *captioning* necesitan reconocer características a nivel de señal como la instrumentación y descriptores de alto nivel como el género. También producen oraciones descriptivas completamente formadas que se asemejan más a las consultas humanas. A través del uso conjunto y procesamiento de información auditiva y lingüística, *music captioning* representa un primer paso hacia el desarrollo de modelos de audio y lenguaje para la comprensión musical. Finalmente, la descripción de música tiene varias aplicaciones directas, como habilitar la búsqueda de música a través de consultas más naturalmente humanas, o proporcionar explicaciones para recomendaciones automáticas de música [46].

El estudio *Audio Retrieval with Natural Language Queries: A Benchmark Study* [2] se centra en la recuperación de eventos de audio (no musicales) mediante consultas en lenguaje natural. Considerando que aprender (en el sentido de *machine learning*) a recuperar audio con consultas en lenguaje natural requiere datos con texto y sonido emparejados, ellos [2] afirman que los datasets de *audio captioning* se prestan naturalmente a esta tarea, ya que contienen audio y una descripción de texto correspondiente al sonido. Proponen aprender *cross-modal embeddings* utilizando datasets de *audio captioning* para el sistema de recuperación. Varios estudios en recuperación de música con consultas en lenguaje natural mencionan esta idea con datasets de *music captioning* [12, 24, 36]. Sin embargo, también se señala que los datasets existentes actualmente no abarcan la diversidad del lenguaje descriptivo del sonido [36]; mientras que algunos estudios terminan utilizando una concatenación de *tags* de diferentes fuentes de anotación como *captions* [12].

2.6. Aprendizaje a partir de Supervisión de Lenguaje

El aprendizaje multimodal todavía ocupa un papel marginal en MIR y aún no ha disfrutado completamente de los beneficios de los modelos de lenguaje modernos [24]. La idea clave detrás de estos modelos es que el lenguaje captura muchas de las abstracciones que los humanos utilizan para navegar por el mundo y, por lo tanto, puede actuar como supervisión para el aprendizaje de propósito general, incluso en tareas que no se basan directamente en el lenguaje.

Los clasificadores generalmente se entrenan para etiquetar ejemplos, con clases predefinidas y fijas. Impulsados por los avances recientes en modelación neuronal de lenguaje y su competencia demostrada en aprendizaje por transferencia, los investigadores han comenzado a explorar menos restrictivas interfaces de lenguaje natural, para acceder a la información categórica subyacente a las señales sin procesar [36]. La mayoría de ese trabajo se ha centrado en el dominio de eventos visuales y de audio, combinando contenido multimedia con subtítulos en lenguaje natural [2, 52, 53, 54].

El éxito de estos esfuerzos depende en gran medida de recursos de entrenamiento a gran escala y de arquitecturas de redes neuronales robustas, que sean lo suficientemente flexibles para modelar la compleja y no monótona relación entre el lenguaje y otras modalidades. En particular, el dominio visual ha obtenido grandes beneficios de la disponibilidad de grandes cantidades de imágenes con descripciones disponibles en la web [52]. Sin embargo, en el dominio general de audio ambiental, pares de audio-descripción a gran escala están menos disponibles y los esfuerzos relacionados se han basado pequeños datasets de *captions* [55, 56].

2.7. *Embeddings* en recuperación de información

Un *embedding* es una representación de elementos en un nuevo espacio, de manera que se preservan las propiedades y relaciones entre los elementos. El objetivo de un *embedding* es generar una representación más simple, donde la simplificación puede implicar una reducción en el número de dimensiones, un aumento en la dispersión de la representación, desentrañar los componentes principales del espacio vectorial, o una combinación de estos objetivos [29].

Los *embeddings* de palabras, de modelos pre-entrenados, se utilizan en diversas aplicaciones, como en la construcción de representaciones para frases, párrafos y documentos [57]. Con modelos de *embedding* de palabras, los documentos preprocesados se mapean a vectores de números reales mediante tecnologías como redes neuronales, reducción de dimensionalidad en la matriz de co-ocurrencia de palabras, entre otras [58].
Dado que tienen en cuenta el contexto donde aparecen las palabras, permite la predicción de palabras faltantes en un documento. En contraste, los motores de búsqueda tradicionales, basados en palabras clave, no pueden resolver el problema de la alta discrepancia entre términos; y considerar las diferencias de significado entre palabras semánticamente similares en el proceso de comparación.

Los modelos de *embeddings* más utilizados/estudiados en la literatura incluyen:

- **Word2vec** [6] (2013): es un conjunto de modelos relacionados que se utilizan para producir *embeddings* de palabras. Estos modelos son redes neuronales superficiales de dos capas que se entrenan para reconstruir contextos lingüísticos de palabras. Word2vec puede utilizar dos arquitecturas de modelo para producir estas representaciones distribuidas de palabras: *Continuous Bag-Of-Words* (CBOW) o *skip-gram* de deslizamiento continuo. En ambas arquitecturas, word2vec considera tanto palabras individuales como una ventana de contexto deslizante a medida que itera sobre el corpus. El problema de la dispersión en word2vec causa que la dimensión de su espacio vectorial sea mayor que otras tecnologías, lo que provoca un uso excesivo de recursos de memoria y una baja robustez [58].

- **GloVe** [7] (2014): acuñado a partir de su nombre en inglés *Global Vectors*, es un algoritmo de aprendizaje no supervisado para obtener representaciones vectoriales de palabras. Resulta en un modelo global de regresión log-bilineal que combina las ventajas de las dos principales familias de modelos en la literatura: factorización de matrices globales y métodos locales de ventana de contexto.

- **BERT** [8] (2019): abreviatura de *Bidirectional Encoder Representations from Transformers*, es un modelo de lenguaje basado en la arquitectura *transformer*, notable por su mejora dramática respecto a anteriores modelos en el estado del arte. El modelo pre-entrenado de BERT puede ser ajustado (*fine-tuning*) con solo una capa de salida adicional, para crear modelos de última generación para una amplia gama de tareas; como respuesta a preguntas e inferencia de lenguaje, sin modificaciones sustanciales en la arquitectura específica de la tarea. *Sentence BERT* (SBERT) [59] es un algoritmo basado en aprendizaje automático que utiliza un *transformer* de oraciones para generar *embeddings* de oraciones mediante una red neuronal siamesa. SBERT puede ser beneficioso para la búsqueda semántica y la similitud textual semántica.

Los sistemas de recuperación de información basados en *embeddings* reciben una entrada del usuario en forma de consulta. Luego, el sistema procesa todas las consultas y genera *embeddings* vectoriales que ayudan a comparar la consulta con la colección de documentos del corpus. Los *embeddings* de la consulta y del documento pueden compararse utilizando

una variedad de métricas de similitud, como la similitud del coseno o el producto punto. Los documentos relevantes se envían al usuario en orden decreciente de relevancia, lo que ayuda a identificar cuáles resultados son mejores.

2.8. Datasets

Para establecer un baseline de la eficacia del prototipo a implementar es necesario evaluarlo con las métricas usuales en SRI y preferentemente en el mismo conjunto de datos que otros trabajos en el tema.

Contrastive Audio-Language Learning for Music [24] entrenaron y evaluaron su modelo de recuperación en un dataset de 250 mil pares (audio, texto) creado a partir de una biblioteca de música en producción. Pero no se encuentra disponible públicamente para utilizarlo y comparar justamente los resultados.

La evaluación de recuperación de música a partir de consultas textuales de *MuLan: A Joint Embedding of Music Audio and Natural Language* [36] fue una colección patentada de 7,000 listas de reproducción curadas por expertos. Que tampoco fue encontrada públicamente.

Dado la idea mencionada en la sección 2.5, se decidió encontrar un dataset de *music captioning*. Sin embargo la mayoría de los utilizados en la literatura son privados. Finalmente se optó por utilizar MusicCaps [60], que fue liberado públicamente por *Google* en una investigación de hace apenas unos meses.

En *HuggingFace* [61] aparece una descripción del dataset, el .csv y un *script* [1] para descargar los clips de YouTube. MusicCaps contiene 5,521 clips de 10 segundos, extraídos de AudioSet [62]. Por cuestiones de conectividad, para esta investigación se utiliza un subconjunto de clips, lo que no causa ningún problema dado que no es necesario tener una cantidad de datos específica con que entrenar. Cada clip contiene una lista de 'aspectos' (*aspect list*) y una descripción textual escrita por músicos.
Una lista de aspectos es, por ejemplo, "*pop, tinny wide hi hats, mellow piano melody, high pitched female vocal melody, sustained pulsating synth lead*".
Mientras que la descripción consta de varias oraciones sobre la música como, "*A low sounding male voice is rapping over a fast paced drums playing a reggaeton beat along with a bass. Something like a guitar is playing the melody along. This recording is of poor audio-quality. In the background a laughter can be noticed. This song may be playing in a bar.* "

[1] https://github.com/nateraw/download-musiccaps-dataset

La escasez de datasets en MIR es un gran obstáculo en el desarrollo del campo. Se debe, en parte, a que a diferencia de otros tipos de multimedia, la música tiende a tener *copyright* más estricto. Lo que impide crear conjuntos de datos con ejemplos representativos, que además sean lo suficientemente grandes para ser útiles en las arquitecturas de redes neuronales utilizadas en el presente para aprender en el campo de texto o imágenes, por ejemplo. Esto ha implicado que varias investigaciones recientes, incluyendo la presente, decidan apoyarse en las capacidades de transferencia de conocimiento que poseen los modelos de lenguaje, como fue explicado en la sección 2.6, para tareas en MIR que incluyan trabajo con texto.

Capítulo 3

Diseño e implementación

3.1. Modelo general

El primer paso consiste en definir qué método utilizar para obtener una representación de la música (sobreponiéndose a la brecha semántica), que pueda ser utilizada para la recuperación semántica con consultas textuales.

Una de las estrategias de recuperación de información (con consultas textuales) utilizada extensamente, como se ha mencionado anteriormente, es mediante metadatos. Se debe recordar que dicha estrategia requiere que las canciones contengan metadatos curados (que no es muy práctico en los escenarios actuales donde la multimedia se genera en grandes cantidades, y libremente). Además, como segunda desventaja, la información conocida sobre las canciones estaría limitada al conjunto fijo de metadatos que se predefinan.

Si se extraen los metadatos (*tags*, *features*) automáticamente con modelos de *Machine Learning* de clasificación y auto-tagging, se hace frente al primer problema mencionado: cómo hacer que todas las canciones tengan metadatos, sin requerir una cantidad de trabajo humano irrealista. Dado que los *tags* se extraen automáticamente, se abre una ventana a contrarrestar la segunda desventaja, ya que la cantidad de tags dependería de la cantidad de modelos de *Machine Learning*. Esto incluso permite que el conjunto no sea fijo. Con el costo de reprocesar la base de datos, se hace posible actualizar y añadir valores.

Otra estrategia, que ha sido analizada por las escasas investigaciones en el tema, es diseñar un modelo para convertir la música al espacio del lenguage natural, o convertir las descripciones textuales y la música a un espacio compartido. El problema fundamental de este acercamiento es que

requiere un conjunto bastante grande de pares texto-música, lo suficientemente diverso y abarcador para maximimar la generalización del modelo. La inexistencia de dichos datasets [13, 36, 24], en conjunto con las dificultades de requerimientos computacionales; no permiten realizar el entrenamiento necesario para implementar esta estrategia.

La propuesta de extraer los *features* automáticamente trabaja bajo la premisa de que el campo de MIR se ha desarrollado en tareas de clasificación y auto-tagging lo suficiente para ser competitivo con metadatos hechos a mano, relativo al esfuerzo que conlleva cada uno.

Para garantizar que el sistema de recuperación tenga en cuenta la similitud semántica, se propone utilizar un aceramiento con *Sentence BERT* (sec. 2.7) para comparar las consultas con las descripciones de la música. Los enfoques tradicionales en SRI para mantener la significación semántica, suelen requerir estructuras complejas y un largo tiempo de preprocesamiento o de ranking (como Latent Semantic Analysis (LSA) [63]). Con el auge de los modelos de lenguaje (alrededor de los últimos de cinco años) y su capacidad de capturar significado semántico y relación contextual; se ha abierto el camino para combinar la eficiencia del modelo vectorial con bolsa de palabras y el aumento en precisión de los modelos como LSA.

Dado que SBERT captura el *embedding* de un texto, y que el *framework* de extracción de *features* lo que devuelve es una lista de *tags*, es necesario convertirla a texto en lenguaje natural.

La tarea de generación de texto a partir de tablas (*table-to-text generation*) consiste en tomar una tabla estructurada como entrada y producir una descripción en lenguaje natural [64]. Tiene buenas perspectivas de aplicación en la comunicación con humanos de manera comprensible y natural, como en la generación de informes financieros, informes médicos, etc.

La mejor alternativa que se encontró para convertir los *features* en texto fue utilizar *TableGPT: few-shot Table-to-text generation with Table Structure Reconstruction and Content Matching*. TableGPT [65] se enfoca en generar texto de alta fidelidad para la generación de texto a partir de tablas con un número limitado de pares de entrenamiento. Abordando la brecha entre la entrada (tablas estructuradas) y la entrada de GPT-2 (lenguaje natural); TableGPT intenta transformar naturalmente las tablas estructuradas en lenguaje natural. Sin embargo, este modelo no se encuentra accesible para ser utilizado directamente con inferencia. Para hacer uso de TableGPT sería necesario recrear todo su proceso de entrenamiento. Pero en el desarrollo de la investigación no se contó con los recursos computacionales para efectuarlo.

Entonces las propuestas factibles son:

- utilizar GPT-2 directamente para generar descripciones a partir de la lista de metadatos
- crear una oración con fuerza bruta, por ejemplo para el género: "The music genre sounds like <*genre classification*>. "
- utilizar un aceramiento similar a MusCALL [24], de concatenar tags en un oración

La primera requiere hacer *prompt engineering* y no garantiza fidelidad con los datos extraídos. Mientras que las otras dos resultarían en descripciones que distan de la forma en que las personas realmente realizan consultas.

Teniendo en cuenta todo esto, se propone intentar las tres estrategias de convertir *tags* en descripciones y comparar los resultados con las tres alternativas.

En resumen, la implementación del prototipo objetivo implica desarrollar una plataforma que permita realizar consultas en lenguaje natural sobre una base de datos de música y obtener resultados con significado semántico. La propuesta del presente trabajo (fig. 3.1) consiste en:

- un sistema de extracción de *features* compuesto por modelos de clasificación de música (incluyendo *features* de bajo y alto nivel).
- las características extraídas, entonces, son convertidas en descripciones utilizando las tres alternativas mencionadas anteriormente, obteniendo tres corpus de *captions*.
- finalmente las descripciones serían procesadas con SBERT para generar los *embeddings* que se utilizan en el sistema de recuperación.

En el apartado de recuperación de información, el sistema consiste en recopilar la consulta del usuario, generar el vector de *embedding* con SBERT y encontrar las canciones más similares comparando el vector con el corpus de *embeddings*, utilizando similitud de coseno. En la imagen 3.2 se observa el diseño del SRI.

3.2. Detalles de implementación

3.2.1. Extracción de *features*

La arquitectura del sistema de extracción de *features* fue relativamente sencilla de implementar. Un sistema de clases que heredan de la clase abstracta *FeaturesExtractor*.

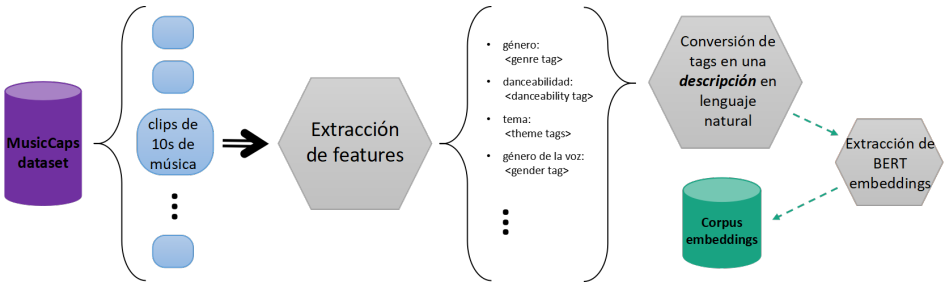


Figura 3.1: Diseño propuesto.

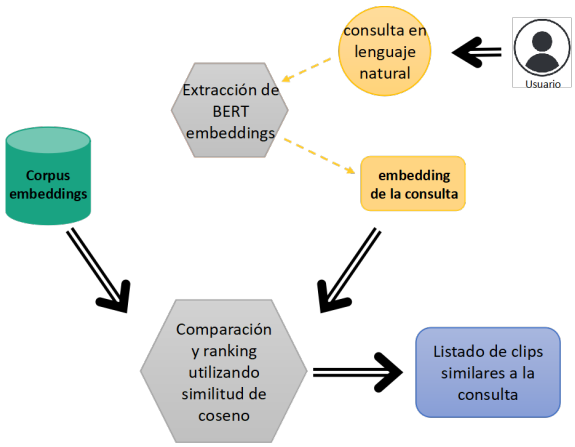


Figura 3.2: Sistema de recuperación utilizando *Sentence BERT embeddings*.

Las propiedades incluyen: una lista de las etiquetas de clase posibles de cada modelo; *feature_description* que retorna una oración en lenguaje natural describiendo una etiqueta de clase; *feature_tags_description* retorna un tag/metadato describiendo una etiqueta de clase. Además de la función *extract_feature*, que recibe un clip de música y retorna el(los) *feature(s)* extraídos con el modelo específico.
Los modelos utilizados, para probar y evaluar el diseño son parte de los que provee la biblioteca *essentia* en conjunto con *tensorflow* [66]. Se utilizaron 9 modelos : Genre Discogs400, MTG-Jamendo genre, Danceability, Mood Happy, Mood Relaxed, Mood Sad, MTG-Jamendo mood and theme, MTG-Jamendo instrument y Voice gender.

Los modelos de clasificación de *essentia* requiren que el audio sea convertido a *embeddings* para recibirlos como entrada, y fue utilizado para ello

el modelo *Discogs-EffNet* [1].

En *musiccaps-subset-feat.csv* [2] (tabla 3.1) se puede observar el subconjunto utilizado de MusicCaps con los *features* extraídos, además de la información originaria del dataset sobre cada clip. Este proceso, en una computadora con 24 GB de RAM y procesador Intel Core i7-4710HQ, tomó alrededor de 18 horas para terminar con los alrededor de 3800 clips.

			Extracted features						
ytid	aspect list	caption	Discogs 400 genre	MTG Jamendo genre	danceability	happy	...	instruments	voice gender
-0Gj8-vB1q4	['low quality', ... , 'ballad']	"The low quality recording ... like something you would hear at Sunday services."	Stage & Screen—Soundtrack	classical	not danceable	non happy	...	['piano', 'violin', 'cello']	female
-0SdAVK79lg	['guitar song', 'piano backing', ... , 'no voice']	"This song features an electric guitar ... This song can be played in a coffee shop."	Blues—Delta Blues	pop	not danceable	non happy	...	['guitar', 'acoustic-guitar', 'bass']	female
.....									

Tabla 3.1: Fragmento del archivo musiccaps-subset-feat.csv

3.2.2. Conversión de tags en una descripción en lenguaje natural

Como fue mencionado en la sección 3.1, la idea era implementar tres alternativas para obtener descripciones a partir de los *features* extraídos anteriormente.

Utilizar GPT-2 con *prompt engineering*

Se intentó unas decenas de prompts utilizando el modelo de completamiento de texto de GPT-2. Sin embargo, no se obtuvo resultados aceptables con ningún de ellos. Teniendo en cuenta dichos resultados se probó con GPT-3.5, desde la página de Open-AI; y esta vez si se observaron buenas descripciones, aunque no completamente fieles, como es de esperar de estos modelos. Sin embargo, no es factible realizar casi 4000 peticiones, ya que a diferencia de GTP-2, GPT-3.5 no es tan accesible. Por lo tanto esta alternativa tuvo que ser descartada.

[1] https://essentia.upf.edu/models.html

[2] https://github.com/NileyGF/Busqueda-semantica-en-audios.-Tesis/blob/main/src/data/musiccaps-subset-feat.csv

Crear una oración con fuerza bruta
Utilizando la propiedad feature_description, se generó una oración por cada *feature* extraído, por cada uno de los clips en el dataset. La concatenación de dichas oraciones en un texto constituyó el corpus de descripciones.
Ejemplos de la metodología para crear las oraciones:

- MTG-Jamendo genre: "The music genre sounds like <*genre classification*>. "
- Mood Relaxed: " Its sound is >*relaxing/not relaxing*>. "
- MTG-Jamendo instrument: "You can hear the sounds of <*comma separated top instruments*>. "
- Voice gender: "There is a <*female/male*>voice. "

Aprovechando la libertad al diseñar el sistema de recomendación, se propuso realizar un segundo corpus de descripciones conteniendo las oraciones separadas además del texto resultado de la concatenación. De forma que este corpus extendido tiene 10 descripciones (que posteriormente son transformadas en *embeddings*) por cada clip (una oración por cada uno de los 9 *features* y el texto completo). Esta idea surge para comparar que ofrece mejores resultados en la recuperación con *embeddings*: una consulta en forma de texto largo u oraciones más cortas, pero repetitivas a través del conjunto de datos.

Concatenar *features* sin intentar que parezca una oración humana
Utilizando la propiedad *feature_tags_description*, se creó una 'oración' por cada clip, donde el valor de cada *feature* es separado del siguiente por un '; '. En algunos, la propiedad modifica ligeramente el valor resultante del modelo de clasificación. Por ejemplo, tiene más sentido semántico utilizar *"{female/male} voice "*, que simplemente *"{female/male}"*.

En *musiccaps-subset-descriptions.csv* [3] (tabla 3.2) se pueden observar las descripciones de cada clip, tanto en formato de oraciones procesadas, como de tags concatenados.

[3]https://github.com/NileyGF/Busqueda-semantica-en-audios.-Tesis/blob/main/src/data/musiccaps-subset-descriptions.csv

			Extracted features				Descriptions	
ytid	aspect list	caption	Discogs 400 genre	MTG Jamendo genre	...	voice gender	description	tags description
-0Gj8-vB1q4	['low quality', ... , 'ballad']	"The low quality recording ... like something you would hear at Sunday services."	Stage & Screen—Soundtrack	classical	...	female	"The music genre is considered to be Stage & Screen, Soundtrack ... There is a female voice. "	"Stage & Screen, Soundtrack; classical; ... ; female voice."
-0SdAVK79lg	['guitar song', 'piano backing', ... , 'no voice']	"This song features an electric guitar ... This song can be played in a coffee shop."	Blues—Delta Blues	pop	...	female	"The music genre is considered to be Blues, Delta Blues ... There is a female voice. "	"Blues, Delta Blues; pop; ... ; female voice."
.....								

Tabla 3.2: Fragmento del archivo musiccaps-subset-descriptions.csv

3.2.3. Extracción de *embeddings* de BERT

Cada uno de los 3 corpus de descripciones fueron posteriormente procesados para crear un vector de *embeddings* utilizando el modelo de la biblioteca *transformers* de *HuggingFace*: **bert-base-uncased**. El proceso consiste en tokenizar (con la función BertTokenizer de *transformers*) el texto:

```
preprocess_tokens = "[CLS]" + text + "[SEP]"
```

Luego, utilizando la biblioteca *torch* y el modelo de BERT se transforma el texto tokenizado en *embeddings*. Al finalizar se guarda como un archivo binario la lista de *embeddings*, que no se encuentran en el repositorio, debido a que ocupan relativamente bastante espacio.

El primer corpus de *embeddings* tomó en computarse, en una computadora con 24 GB de RAM y procesador Intel Core i7-4710HQ, alrededor de 2 horas. El segundo, que consiste en una extensión del primero con 10 veces la cantidad de *embeddings*, tomó entre 14 y 15 horas de procesamiento. El tercer corpus, constituido por la concatenación de tags tomó entre 1 y 2 horas en terminar.

Como último detalle, en esta parte de la implementación, al finalizar de procesar cada corpus, se creó un diccionario que establece por cada *embedding* a que clip corresponde. Esto es particularmente importante en el caso del segundo, ya que existen 10 *embeddings* por cada clip.

3.2.4. Recuperación de Información. Comparación y ranking

El método de obtener la música más cercana, dado una consulta, consiste en obtener un vector de *embeddings*, utilizando el mismo proceso descrito anteriormente en 3.2.3, luego se compara dicho vector con cada uno de los *embeddings* en el corpus (solo se emplea uno de los tres), utilizando la similitud del coseno entre el ángulo de los vectores. Finalmente se ordenan descendentemente, ya que el coseno es mayor entre vectores más cercanos. Se puede especificar cuantos resultados obtener: todos, o solo los k más similares a la consulta.

Después de ordenar los *embeddings* se busca la correspondencia entre ellos y los clips, de forma que la relevancia de un clip es la relevancia del primer *embedding* correspondiente a él (o sea, el de mayor valor de coseno). Esto es particularmente importante para el segundo corpus y cualquier otro que se pueda diseñar eventualmente donde a un clip le corresponda más de un vector de *embeddings* (por ejemplo, si la descripción contiene más de 512 *tokens* no puede ser procesada con BERT, y puede solucionarse dividiéndola en varias subdescripciones más cortas).

Capítulo 4

Breve Experimentación

4.1. Consideraciones generales

Para evaluar un SRI, tradicionalmente, se parte de un conjunto de documentos y un conjunto de consultas con una relación de relevancia establecida entre ambos [4]. Esta relación es fundamental para identificar los elementos relevantes para una consulta, e incluso puede determinar el grado de relevancia. Algunos ejemplos de datasets para evaluar SRIs son: *antique*, *car*, *cranfield*, *msmarco-passage*, *nfcorpus*, *nyt*.

Las estrategias de evaluación centradas en el usuario buscan tener en cuenta diferentes factores en la percepción de las cualidades musicales, en particular de la similitud musical. Esto es especialmente importante ya que las nociones de similitud musical están probremente definidas. El acuerdo entre los humanos sobre el parecido entre dos piezas musicales está limitado a alrededor del 80 % según se afirma en la literatura [14].
En general los SRIs evalúan la semejanza utilizando la métrica de similitud de coseno [57]. Esta técnica calcula la similitud entre dos elementos basándose en el ángulo entre sus representaciones vectoriales. Su popularidad se basa en que es invariante a la magnitud del vector, es eficiente de computar y que en epacios multi-dimensionales captura la orientación de los datos.

Un gran número de estudios de recuperación de información han demostrado que los usuarios de los sistemas de recuperación tienden a prestar atención, principalmente, a los primeros resultados [29]. Por lo tanto, las métricas de recuperación de información se centran en comparaciones basadas en los primeros resultados recuperados. Estas métricas generalmente se calculan en una posición, digamos k, y luego se promedian sobre todas las consultas.

El hecho de que no existen datasets de recuperación de información de música implica que en todos los experimentos en el tema se definen las consultas y la relevancia entre una consulta y la música de forma diversa.

Utilizando la información proveída por el dataset (sec. 2.8) fueron concebidos dos conjuntos de consultas. El primero con las descripciones creadas por músicos y el segundo con la lista de aspectos (convertida en oraciones concatenando los *tags*). Durante el preprocesamiento se extraen los *embeddings* de ambos conjuntos (utilizando SBERT [59]) y se guardan en archivos binarios.
Es necesario establecer una relación de relevancia entre consultas y clips. Para ello fue diseñado un algoritmo que itera por los *embeddings* de cada conjunto de consultas.
El algoritmo crea una lista *R*, donde para la consulta i-ésima:

```
R[i] = [(song_id_1, cosine_similarity_i1), ... ,(song-id_j,
    cosine_similarity_ij)]
```

R[i] contiene una tupla por cada consulta *j* que cumpla:

$$cosine\_similarity(i,j) \geq 0{,}95$$

ya que se considera que a descripciones similares corresponden clips similares. De esta forma siempre se obtiene el clip que describe la consulta i pues $cosine\_similarity(i,i) = 1$.

En *Toward Universal Text-To-Music Retrieval* [12] se realiza uno de los análisis más recientes y abarcadores sobre modelos de recuperación de música utilizando texto. Ellos utilizan un subconjunto fijo de 1000 consultas para evaluar varios modelos; por lo que este trabajo considera mantener ese número.

Para ello se extrae un subconjunto aleatorio de 1000 consultas antes de cada proceso de evaluación. Se utilizó un *seed* para que al ejecutar el proceso de evaluación se obtengan resultados consistentes.

Debido a que fueron propuestos 2 conjuntos de consultas y 3 de descripciones (sec. 3.2.2), cada experimento debe ser repetido 6 veces. El nombre de cada combinación se define como: $consultas - descripciones$. Más detalles en la figura 4.1.

4.2. Recobrado y precisión

El primer experimento evalúa el recobrado (*recall*) y la precisión promedio (*mean average precision* mAP). El recobrado representa la proporción

```
descripciones = {corpus: "Descripciones creadas generando
    oraciones a partir de feature_description y uniendolas en un
    texto descriptivo.",
                ext_corpus: "Extension del primer corpus;
    conteniene los mismos textos y las oraciones que lo componen
    separadas.",
                tag_corpus: "Consiste basicamente en la
    concatenacion de todos los features que retornan los modelos
    de la seccion \ref{subsec:essentia} por cada clip."}
consultas = {capt: "Columna \'caption\' del dataset MusicCaps, son
     descripciones textuales escritas por musicos.",
             aspect: "Esta conformado por la concatenacion de las
    etiquetas de la columna \'aspect list\' del dataset MusicCaps.
    "}
```

Figura 4.1: Definición de los conjuntos de descripciones y consultas.

entre las canciones recuperadas que son relevantes y el número total de canciones relevantes. En particular se calcula el *recall@k* con k =1, 5, 10, 50; que solo considera los primeros k elementos recuperados. El mAP promedia los resultados de precisión en cada rango donde se encuentra un clip relevante [67]. La precisión representa la fracción de elementos recuperados que son relevantes para la consulta del usuario. Esta métrica se ha utilizado durante mucho tiempo como el "estándar de oro" de facto para la evaluación de sistemas de recuperación [68]. Se calcula el mAP general y mAP@10 (donde se consideran solo los top 10 clips recuperados).

Todas las métricas se calculan para cada consulta y luego se promedian resultando en los valores que se muestran en la tabla 4.1.

	R@1	R@5	R@10	R@50	mAP@10	mAP
capt-corpus	0.00082	0.00499	0.0096	0.03492	0.107	0.0352
capt-ext_corpus	0.00076	0.0059	0.0095	0.03485	0.101	0.0357
capt-tag_corpus	0.00025	0.00202	0.0057	0.0273	0.075	0.0374
aspect-corpus	0.00124	0.0047	0.0117	0.0424	**0.124**	0.0296
aspect-ext_corpus	**0.00137**	0.0053	0.0098	0.0344	0.078	0.028
aspect-tag_corpus	0.00056	**0.00761**	**0.01254**	**0.0524**	0.096	**0.0381**

Tabla 4.1: Resultados de las 6 combinaciones consultas-corpus respecto a recall y mean average precision.

Se observa que, en general, utilizar el *aspect list* como consultas provee mejores resultados, en particular cuando se utiliza como corpus los *tags* concatenados. Además el mAP, que se supone que es muy representati-

vo, tiene mejores valores que el recobrado. En cuanto a las consultas que consisten en las descripciones creadas por expertos, los mejores valores se obtienen con el corpus conformado por un texto descriptivo por cada clip.

También se ilustra que tanto el par de consultas-de-texto-descriptivo, corpus-de-texto-descriptivo; como el par de consultas-de-tags-concatenados, corpus-de-tags-concatenados; resultaron en los mejores valores. Esto puede indicar que es mejor que el corpus a utilizar para describir la música tenga la forma que más prevalece en las consultas de los usuarios.

4.3. Índice de aciertos

El segundo experimento evalúa una métrica no muy común, pero que refleja de forma simple un factor importante para los usuarios. Fue calculado el índice del modelo de mostrar elementos relevantes en los primeros k clips recuperados (para k que refleja cuantos resultados usualmente comprueban los usuarios (1, 5, 10)).
Entonces el *hits_rate@k* para una consulta tiene como valor 1 si en los primeros k resultados hay algún clip relevante y 0 si no. Finalmente se promedia, para un k fijo, todos los *hits_rate@k*, obteniendo el índice de consultas donde en los top k resultados se recuperan buenos clips.

	hits_rate@1	hits_rate@5	hits_rate@10
capt-corpus	0.052	0.193	**0.333**
capt-ext_corpus	0.05	0.182	0.314
capt-tag_corpus	0.027	0.142	0.248
aspect-corpus	**0.087**	**0.215**	0.301
aspect-ext_corpus	0.052	0.137	0.209
aspect-tag_corpus	0.037	0.18	0.303

Tabla 4.2: Resultados de las 6 combinaciones consultas-corpus respecto al índice de aciertos.

En la tabla 4.2 se observa que los resultados para esta métrica son bastante uniformes y los mejores valores se obtienen con el corpus número 1. Con la lista de aspectos como consultas y el primer corpus se obtiene casi el doble de resultados relevantes en la primera posición que con el resto.
Además, atendiendo a la motivación de este experimento se obtuvo que aproximadamente un tercio de las consultas tienen un clip relevante entre los primeros 10 (de más de 3800 clips).

4.4. Discusión

Se pudo observar que, en ambos experimentos, no se obtienen resultados muy distintos entre *corpus* y *ext_corpus*. Lo que conlleva a concluir que del texto se infiere aproximadamente la misma información semántica que de las oraciones que lo componen. El hecho de procesar un corpus 9 veces más grande (en este caso porque solo se utilizaron 9 modelos) aumentó el tiempo de extracción de *embeddings* y el de recuperación considerablemente. Por lo que se debe considerar que no es una buena estrategia aumentar el corpus como se propuso.

Antes de considerar comparar los resultados obtenidos con otras investigaciones hay que tener en cuenta que ninguna utliza el mismo conjunto de música ni de consultas. Tampoco se comparte la estrategia de recuperación ya que los modelos de MuLan [36], MusCALL [24] y Contrastive Sentence BERT [12] utilizan modelos de aprendizaje contrastivo. Debido a lo novedoso de la estrategia propuesta en este trabajo y al problema que existe en el campo de recuperación de música a partir de texto respecto a la falta de datasets y la ausencia de un protocolo uniforme para la evaluación multimodal, no es factible hacer una comparación que lance alguna conclusión significativa ya que no hay puntos comunes. Sin embargo se espera que, con la publicación de MusicCaps [60] y otros datasets en el futuro, surjan nuevas investigaciones o se reevaluen modelos existentes con métricas y datos comunes.

Conclusiones

Esta tesis presentó un enfoque para la recuperación de música con consultas en lenguaje natural, que aprovecha el poder de los clasificadores de MIR y los *embeddings* de oraciones.

A través de la exploración, se ha mostrado el potencial de utilizar clasificadores de aprendizaje automático (*machine learning*) para extraer características matizadas de la música, en un intento de traducir sus cualidades auditivas en una descripción en lenguaje natural. Este paso representa un puente crítico entre la naturaleza abstracta de la música y el ámbito lingüístico de la comunicación humana, reduciendo la brecha semántica y allanando el camino para mejores sistemas de recuperación de música. Ahí radica la importancia de continuar por esta vertiente de investigación y encontrar otras formas de realizar la transformación de las características musicales a textos de que representen la percepción humana. Es probable que una forma de mejorar en este esfuerzo sea siguiendo los grandes avances en modelos de lenguaje de los últimos años (ya sea TableGPT u otro enfoque).

Es importante reconocer las limitaciones del trabajo. Aunque el prototipo es prometedor, todavía existen desafíos por abordar, como la interpretabilidad de las descripciones en lenguaje natural y la escalabilidad del proceso de recuperación. También es importante señalar que aún no existen datasets para la tarea de recuperación de texto a música, y los datasets adaptables para la tarea no son de un tamaño adecuado. Se reconoce que las causas de la indisponibilidad de datasets también deben abordarse. Entre ellas se encuentran: el problema mencionado de los derechos de autor (*copyright*), el hecho de que la alineación de texto y audio es una tarea computacional difícil que debe ser supervisada por humanos, y la ausencia de un estándar ampliamente adoptado para la representación multimodal de la música. Enfoques recientes de aprendizaje de representaciones multimodales han mostrado avances en muchos dominios al aprovechar enormes datos de la web, pero no en dominios musicales (al menos no en investigaciones públicas).

El potencial para futuros avances en la recuperación de música y sistemas de recomendación es vasto. Se espera que esta investigación siente las bases para estos esfuerzos futuros. Este es un primer intento de construir una interfaz de lenguaje natural de forma libre para música, y hay mucho espacio para mejorar.

Recomendaciones

Para futuras investigaciones se recomienda utilizar TableGPT [65] u otro modelo de *table-to-text generation* para la tarea de obtener una descripción de la música a partir de una lista de features y comparar el resultado con los presentados en este trabajo, que utilizan un acercamiento de fuerza bruta.
Otra idea a incorporar, que debería resultar en un prototipo más eficiente, es aumentar la cantidad y diversidad de modelos de clasificación en el sistema de extracción de features (sec. 3.2.1).

En la sección de discusión de los resultados de la experimentación (sec. 4.4), se plantea que *ext_corpus* no es una alternativa factible para mejorar las descripcioes; sin embargo se podría evaluar un corpus conformado por un par de vectores de *embeddings* por cada canción: uno de descripciones en forma de texto y otro formado por tags (o sea, una combinación de las ideas del primer y el tercer corpus propuesto). Además, en el experimento 1 (sec. 4.2) se analiza que los resultados mejoran cuando las consultas tienen la misma forma que las descripciones. Un corpus mixto puede ser una solución robusta al hecho de que no se puede preveer que tipo de consulta utilizarán los usuarios en la búsqueda.

Para finalizar se recomienda evaluar el prototipo realizando estudios de ablación para comprobar el impacto de cada parte del sistema. Por ejemplo, cuánto mejoran los resultados al incluir la búsqueda utilizando *embeddings* de BERT en comparación con el modelo vectorial (de bolsa de palabras) tradicional.

Bibliografía

[1] Hannah Ritchie, Edouard Mathieu, Max Roser, and Esteban Ortiz-Ospina. Internet. *Our World in Data*, 2023. https://ourworldindata.org/internet. (Citado en la página 1).

[2] A. Sophia Koepke, Andreea-Maria Oncescu, João F. Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries: A benchmark study. *IEEE Transactions on Multimedia*, 25:2675–2685, 2021. (Citado en las páginas 1, 12 y 13).

[3] Bayan Alabdullah, Natalia Beloff, and Martin White. Rise of big data – issues and challenges. In *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pages 1–6, 2018. (Citado en la página 1).

[4] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008. (Citado en las páginas 1 y 25).

[5] Carlos Fleitas Aparicio and Marcel E. Sanchez Aguilar. Conf, 2021. (Citado en la página 2).

[6] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. (Citado en las páginas 2 y 14).

[7] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, 2014. (Citado en las páginas 2 y 14).

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language un-

derstanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. (Citado en las páginas 2 y 14).

[9] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018. (Citado en la página 2).

[10] Shay Palachy Affek. Document embedding techniques, 2019. Accessed on October, 2023. (Citado en la página 2).

[11] Jian Zhang. Lecture 5: Multimedia information retrieval, 2007. (Citado en la página 3).

[12] Seungheon Doh, Minz Won, Keunwoo Choi, and Juhan Nam. Toward universal text-to-music retrieval. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2022. (Citado en las páginas 3, 9, 10, 12, 26 y 29).

[13] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. Multimodal music information processing and retrieval: Survey and future challenges. *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 10–18, 2019. (Citado en las páginas 4, 6, 8 y 18).

[14] Markus Schedl, Emilia Gómez, and Julián Urbano. Music information retrieval: Recent developments and applications. *Found. Trends Inf. Retr.*, 8:127–261, 2014. (Citado en las páginas 6, 7, 10 y 25).

[15] Francesc Alías, Joan Claudi Socoró, and Xavier Sevillano. A review of physical and perceptual feature extraction techniques for speech, music and environmental sounds. *Applied Sciences*, 6:143, 2016. (Citado en la página 8).

[16] Youngmoo E. Kim, Erik M. Schmidt, Raymond Migneco, Brandon G. Morton, Patrick Richardson, Jeffrey J. Scott, Jacquelin A. Speck, and Douglas Turnbull. State of the art report: Music emotion recognition: A state of the art review. In *International Society for Music Information Retrieval Conference*, 2010. (Citado en la página 8).

[17] Safaa Allamy and Alessandro Lameiras Koerich. 1d cnn architectures for music genre classification. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 01–07, 2021. (Citado en la página 8).

[18] M AthulyaK and S. Sindhu. Deep learning based music genre classification using spectrogram. *Social Science Research Network*, 2021. (Citado en la página 8).

[19] Lvyang Qiu, Shuyu Li, and Yunsick Sung. Dbtmpe: Deep bidirectional transformers-based masked predictive encoder approach for music genre classification. *Mathematics*, 2021. (Citado en la página 8).

[20] Quazi Ghulam Rafi, Mohammed Noman, Sadia Zahin Prodhan, Sabrina Shajeen Alam, and Dipannyta Nandi. Comparative analysis of three improved deep learning architectures for music genre classification. *International Journal of Information Technology and Computer Science*, 13:1–14, 2021. (Citado en la página 8).

[21] Shweta Koparde, Vaishnavi R Bhadgaonkar, Kalyani N Patil, Gauri N Basutkar, and Dhanashri D Gayke. A survey on music genre classification using machine learning. 2021. (Citado en la página 8).

[22] Ndiatenda Ndou, Ritesh Ajoodha, and Ashwini Jadhav. Music genre classification: A review of deep-learning and traditional machine-learning approaches. *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6, 2021. (Citado en la página 8).

[23] Shajin Prince, Justin Jojy Thomas, Sharon Jostana J, Kakarla Preethi Priya, and J Joshua Daniel. Music genre classification using deep learning - a review. *2022 6th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, pages 1–5, 2022. (Citado en la página 8).

[24] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Contrastive audio-language learning for music. *ArXiv*, abs/2208.12208, 2022. (Citado en las páginas 9, 10, 11, 12, 15, 18, 19 y 29).

[25] Kosetsu Tsukuda, Keisuke Ishida, and Masataka Goto. Lyric jumper: A lyrics-based music exploratory web service by modeling lyrics generative process. In *International Society for Music Information Retrieval Conference*, 2017. (Citado en la página 9).

[26] Jongpil Lee, Nicholas J. Bryan, Justin Salamon, Zeyu Jin, and Juhan Nam. Disentangled multidimensional metric learning for music similarity. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics,*

Speech and Signal Processing (ICASSP), pages 6–10, 2020. (Citado en la página 9).

[27] Bochen Li and Aparna Kumar. Query by video: Cross-modal music retrieval. In *International Society for Music Information Retrieval Conference*, 2019. (Citado en la página 9).

[28] Parth Patel. Music retrieval system using query-by-humming. 2021. (Citado en la página 9).

[29] Bhaskar Mitra and Nick Craswell. Neural models for information retrieval. *ArXiv*, abs/1705.01509, 2017. (Citado en las páginas 9, 10, 13 y 25).

[30] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521:436–444, 2015. (Citado en las páginas 9 y 10).

[31] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. (Citado en la página 10).

[32] Li Deng and Dong Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7:197–387, 2014. (Citado en la página 10).

[33] Geoffrey E. Hinton, Liya Deng, Dong Yu, George E. Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew W. Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29:82–97, 2012. (Citado en la página 10).

[34] Michael J. Taylor, Hugo Zaragoza, Nick Craswell, Stephen E. Robertson, and Christopher J. C. Burges. Optimisation methods for ranking functions with multiple parameters. In *International Conference on Information and Knowledge Management*, 2006. (Citado en la página 10).

[35] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. Learning to match using local and distributed representations of text for web search. *Proceedings of the 26th International Conference on World Wide Web*, 2016. (Citado en la página 10).

[36] Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. Mulan: A joint embedding of music audio

and natural language. In *International Society for Music Information Retrieval Conference*, 2022. (Citado en las páginas 10, 12, 13, 15, 18 y 29).

[37] Ja-Hwung Su, Tzung-Pei Hong, Yu-Tang Chen, and Chu-Yu Chin. High-performance content-based music retrieval via automated navigation and semantic features. *Eng. Appl. Artif. Intell.*, 115:105267, 2022. (Citado en la página 10).

[38] Òscar Celma, Perfecto Herrera, and Xavier Serra. Bridging the music semantic gap. 2006. (Citado en la página 10).

[39] Òscar Celma, Perfecto Herrera, and Xavier Serra. A multimodal approach to bridge the music semantic gap. In *International Conference on Semantics and Digital Media Technologies*, 2006. (Citado en la página 10).

[40] Sergio Oramas, Luis Espinosa-Anke, and Shuo Zhang. Natural language processing for mir, 2016. Accessed on November, 2023. (Citado en la página 11).

[41] Sergio Oramas, Luis Espinosa Anke, Mohamed Sordo, Horacio Saggion, and Xavier Serra. Information extraction for knowledge base construction in the music domain. *Data Knowl. Eng.*, 106:70–83, 2016. (Citado en la página 11).

[42] Sergio Oramas, Mohamed Sordo, Luis Espinosa Anke, and Xavier Serra. A semantic-based approach for artist similarity. In *International Society for Music Information Retrieval Conference*, 2015. (Citado en la página 11).

[43] Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. Multimodal deep learning for music genre classification. *Trans. Int. Soc. Music. Inf. Retr.*, 1:4–21, 2018. (Citado en la página 11).

[44] Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. Music mood detection based on audio and lyrics with deep neural net. *ArXiv*, abs/1809.07276, 2018. (Citado en la página 11).

[45] Byungsoo Jeon, Chanju Kim, Adrian Kim, Dongwon Kim, Jangyeon Park, and Jung-Woo Ha. Music emotion recognition via end-to-end multimodal neural networks. In *RecSys Posters*, 2017. (Citado en la página 11).

[46] Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Muscaps: Generating captions for music audio. *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. (Citado en las páginas 11 y 12).

[47] Tianuhi Cai, Michael Mandel, and Di He. Music autotagging as captioning. In *NLP4MUSA*, 2020. (Citado en la página 11).

[48] Brian Whitman and Ryan M. Rifkin. Musical query-by-description as a multiclass learning problem. *2002 IEEE Workshop on Multimedia Signal Processing.*, pages 153–156, 2002. (Citado en la página 11).

[49] Keunwoo Choi, György Fazekas, and Mark B. Sandler. Automatic tagging using deep convolutional neural networks. In *International Society for Music Information Retrieval Conference*, 2016. (Citado en la página 12).

[50] Jongpil Lee, Jiyoung Park, Keunhyoung Luke Kim, and Juhan Nam. Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms. *ArXiv*, abs/1703.01789, 2017. (Citado en la página 12).

[51] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik M. Schmidt, Andreas F. Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. *ArXiv*, abs/1711.02520, 2017. (Citado en la página 12).

[52] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, 2021. (Citado en la página 13).

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021. (Citado en la página 13).

[54] Arsha Nagrani, Paul Hongsuck Seo, Bryan Seybold, Anja Hauth, Santiago Manén, Chen Sun, and Cordelia Schmid. Learning audio-video modalities from image captions. In *European Conference on Computer Vision*, 2022. (Citado en la página 13).

[55] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an audio captioning dataset. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740, 2019. (Citado en la página 13).

[56] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *North American Chapter of the Association for Computational Linguistics*, 2019. (Citado en la página 13).

[57] J. Brundha and K. N. Meera. Vector model based information retrieval system with word embedding transformation. *2022 10th International Conference on Emerging Trends in Engineering and Technology - Signal and Information Processing (ICETET-SIP-22)*, pages 01–04, 2022. (Citado en las páginas 13 y 25).

[58] Ye Yuan. Improving information retrieval by semantic embedding. 2020. (Citado en las páginas 13 y 14).

[59] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*, 2019. (Citado en las páginas 14 y 26).

[60] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matthew Sharifi, Neil Zeghidour, and Christian Havnø Frank. Musiclm: Generating music from text. *ArXiv*, abs/2301.11325, 2023. (Citado en las páginas 15 y 29).

[61] Dataset card for musiccaps, 2023. Accessed on September, 2023. (Citado en la página 15).

[62] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017. (Citado en la página 15).

[63] Peter W. Foltz. Latent semantic analysis for text-based research. *Behavior Research Methods, Instruments, & Computers*, 28:197–202, 1996. (Citado en la página 18).

[64] Yang Yang, Juan Cao, Yujun Wen, and Pengzhou Zhang. Table to text generation with accurate content copying. *Scientific Reports*, 11, 2021. (Citado en la página 18).

[65] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *International Conference on Computational Linguistics*, 2020. (Citado en las páginas 18 y 32).

[66] Pablo Alonso-Jiménez, Dmitry Bogdanov, Jordi Pons, and Xavier Serra. Tensorflow audio models in Essentia. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. (Citado en la página 20).

[67] Ajitesh Kumar. Mean average precision (map) for information retrieval systems, 2023. Accessed on December, 2023. (Citado en la página 27).

[68] Steven M. Beitzel, Eric C. Jensen, and Ophir Frieder. *Encyclopedia of Database Systems*, chapter MAP, pages 1691–1692. Springer US, Boston, MA, 2009. (Citado en la página 27).