Master in Sound and Music Computing

Universitat Pompeu Fabra

Semantic Sound Similarity with Deep Embeddings for Freesound

Recep Oğuz Araz

Co-Supervisor: Dmitry Bogdanov

Co-Supervisor: Pablo Alonso

Co-Supervisor: Frederic Font

August 2023

Master in Sound and Music Computing

Universitat Pompeu Fabra

Semantic Sound Similarity with Deep Embeddings for Freesound

Recep Oğuz Araz

Co-Supervisor: Dmitry Bogdanov

Co-Supervisor: Pablo Alonso

Co-Supervisor: Frederic Font

August 2023

Dedication

I would like to dedicate this work to my:

Grandpa, my first mentor, Mehmet Recep Çöl, for teaching me how to interact with any kind of system: mechanical, electrical, or biological.

Grandma, Fatma Çöl, for teaching me how to always be on the lookout.

Mom, Nilgün Çöl, for leading me by example with her research and demonstrating that there is no useless subject.

Sister, Elif Naz Araz, and brother, Mustafa Okan Araz, for their endless support in life.

Although the following wonderful people are not a part of my family by blood, I also dedicate this work to them. My lifelong friend, Ayhan Alp Aydeniz, who made it difficult to summarize my gratitude for him with a single sentence; my exemplary friend Haldun Balım, for setting the bar so high in life; my dear friend Carlos Cedeño, for sharing a life through electronic music; and Ilse Meijer, whose unconditional love and support has made my recent endeavors possible.

Acknowledgements

I would like to thank my supervisors Dmitry Bogdanov and Pablo Alonso, and Frederic Font for taking the time to question and extend my research. I am grateful to them for the extensive help that they have provided and for getting as excited as I got with the results of this work.

I also would like to thank Alastair Porter for collaborating on technical aspects; Xavier Favory for providing ideas, and reviewing my work; Perfecto Herrera for a fruitful discussion on the properties of sounds; Xavier Serra, Barış Bozkurt, and Furkan Yeşiler for their academic guidance; Ryan Groves, Maarten Grachten, Joan Serrà, Santiago Pascual, and Jordi Pons for extending my knowledge of sound, signal processing, and machine learning through the internships that I conducted alongside this degree.

Finally, I should express how pleased I am with my classmates for being involved in various subjects with me, especially with Christos Plachouras for being a companion throughout my pursuit in the field of music information retrieval. Of course, we not only worked towards a degree together but also enjoyed what Barcelona has to offer.

Abstract

Freesound is an online platform where people using sounds for various purposes can share or download audio clips. In such platforms, it is crucial that the users are provided with accurate sound recommendations, which becomes challenging due to the large size of the audio collection, complexity of the sound properties, and the human aspect of the recommendations. To provide sound recommendations, Freesound features a "similar sounds" function. However, this function primarily relies on creating a digital representation of audio clips that assesses the acoustic characteristics of sounds, which proves to be insufficient for accurately capturing their semantic properties. This limitation reduces the content-based retrieval capabilities of Freesound users. Moreover, the audio representation is created by hand-picking features that were engineered using domain knowledge. Today, in various fields related to audio, this approach has been replaced by using neural networks as feature extractors. In this work, we search for pretrained general-purpose neural networks that can be used to represent the semantic content of audio clips. We choose 8 such models and compare their semantic sound similarity performances both objectively and subjectively. During the integration of deep embeddings in the sound similarity system, we explore numerous design choices and share valuable insights. We use the FSD50K evaluation set for all experiments and report various objective metrics using the sound class hierarchy to perform multi-level analysis, including class- and family-level. We find out that most of the neural networks outperform the hand-made representation subjectively and objectively. Specifically, the multi-modal representation learning model CLAP that uses natural language and audio as modalities outperforms other models by a significant margin, while the models that attempt to leverage the CLIP model for creating tri-modal representations fail.

Keywords: Freesound; Sound Similarity; Sound Semantics; Content-based Retrieval; Neural Networks; Deep Embeddings; Multimodal Representation Learning; Representation Learning; Audio Information Retrieval; Recommendation Systems

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Sound is a crucial part of various types of productions. It can create an immersive sensation by itself, or it can complement other senses. Therefore, a large number of professionals and hobbyists work with audio on a daily basis, using countless types of sounds in their productions. In order to access sound collections, platforms such as Freesound [1], Splice[2], or Epidemic Sound[3] are used regularly. The large amount of audio content on these platforms poses challenges to storing sounds in an organized manner [1], [2], [3]. To facilitate access, it is crucial for such audio collections to be accompanied by technologies such as search algorithms, automatic classification systems, or recommendation systems. For instance, Freesound is equipped with a text-based search engine and a *similar sounds* function to help users browse through its collection.

The sound similarity function provides sound recommendations based on similarity relations between sounds. It provides users with similar sound recommendations to what they are looking for. Due to the complex nature of sounds, the broad range of sound sources in the real world, and the human aspect of recommendations; it

[1] https://freesound.org/
[2] https://splice.com/
[3] https://www.epidemicsound.com/

is challenging for sound similarity systems to recommend relevant sounds to users. Various propositions have been made to solve this challenge.

In the early stages of sound similarity systems, the primary objective was to develop computational methods that could effectively model the semantic content of sounds. Many early efforts focused on crafting features that were acoustically and perceptually motivated [4], [5], [6], [7]. However, it became evident that this approach had limitations, as exemplified by the performance of Freesound's sound similarity function, which follows a similar methodology.

In contrast, like in many other fields, data-driven methods such as deep learning have been replacing hand-crafted features in sound similarity functions [8], [9], [10]. Moreover, through the use of large collections of audio, general-purpose deep learning methods are able to learn sound semantics in the form of broad sound classes [11], [12], [13]. However, it's important to note that these approaches have not yet achieved human-level capabilities in terms of capturing fine-grained sound content details. Recent research has highlighted their limitations in effectively incorporating natural language or understanding the temporal sequencing of events in audio recordings [14].

Although being limited in the semantic capabilities, the improvements in deep learning based methods for semantic sound modeling prompt an investigation of their performance on Freesound. In order to test the semantic sound modeling capabilities of deep learning methods and to improve the sound similarity function of Freesound semantically, we explore various design choices for creating sound similarity systems and a variety of deep learning models.

1.2 Key concepts

In this section, we define the key concepts surrounding our work.

1.2.1 Sound similarity

The similarity of sounds is based on the relationships between their properties. Therefore, we begin by defining sound properties and continue by defining possible relationships. Sound waves are pressure fluctuations propagating through a media such as air, that are created by a vibrating physical process. As a result, sound waves have qualities that can be attributed to the source that created them, the state that the source was in, the physical process that forced the source to vibrate, the media that the sound propagates in, and the distribution of other objects in the surroundings of the source. For example, a tree falling down in a forest because of natural force will create a sound. This sound can be characterized by the mass and the type of the falling tree (source); the strength and duration of the natural force (force); the position and the material of the objects in the forest (environment); and the density and chemical compounds of the air molecules surrounding them (propagation media). Changing any part of these components will create a distinct sound, regardless of the existence of a perceiver.

Whether there is no one around to hear it, or there is a living being, the physical properties of sound such as amplitude, duration, or frequency components remain the same. These physical properties are therefore called the acoustic properties of the sound. Since we are interested in how humans describe these properties, we need to take into account the human perspective. Just like all the other senses such as vision, taste, or touch; sound has a perceptive and cognitive component.

An example of how the tree falling down sound can be perceived by a human could be "The tree cracked very loud for a brief period". Loudness and length are properties that can be perceived differently by different people depending on their age or genetics. These sound properties that are based on perception are hence called perceptual properties of sound. Perceptual properties result from acoustic processes: a person calling a sound "very loud" is the result of the sound wave's frequency distribution and the amplitudes of the individual frequencies; and "brief period" is a result of the actual duration of the sound source vibrating, and how long it excites

the auditory system.

Other descriptions of the tree falling down incident could be: "a tall, plane tree falling down because of a landslide" (a prevalent tree in Gaziantep, Istanbul, and Barcelona) or "a tree falling down, creating a thick thump sound". These are descriptions of what the sound *means* to observers, based on what they think about how the sound was produced, going into detail about the source and the force that created the sound. These descriptions require a cognitive process that maps the sound perception to a physical process, which is called the semantics of the sound. The semantic content of a sound can be described in a multitude of detail levels. It can be as broad as a "thump sound" or it can contain more details as in the descriptions given at the beginning of this paragraph. As discussed previously, the semantic capabilities of current sound modeling systems have not yet reached the human-level. Therefore, in this work we focus on semantic modeling capabilities in terms of broad, simple descriptions. More discussion on simple audio semantics is provided in Section 2.1.

Figure 1 displays the relationship between sound properties using a Venn diagram. It is meant to model the dependencies between the sound properties.

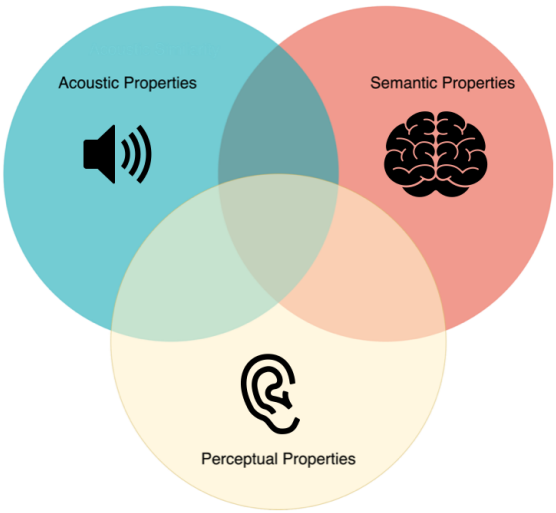

Figure 1: Acoustic, perceptual, and semantic properties representation.

Now that we defined the properties of sounds, we can define the different types of sound similarity. Acoustic similarity is sharing acoustic properties; perceptual

similarity, perceptual properties; and semantic similarity is defined in the same manner. A movie maker may look for multiple examples of "the sound of a woman walking on high heels in a hall", then the movie maker will need semantic similarity. A music producer using a loop library may like how a synthesizer sounds but may need a different melody, then he/she will need perceptually similar but semantically different sounds. A Music Information Retrieval researcher may need a musical instrument that has white-noise-like properties to use for data augmentation, then it will need acoustically similar sounds. Freesound is used for a wide range of purposes, including these tasks. Ideally, a user should be able to choose what type of similarity they are interested in. However, this would require more computation and storage, both of which are limited resources.

1.2.2 Audio Information Retrieval systems

Audio Information Retrieval (AIR) systems help users search for content within a collection of items such as music, speech, and environmental sound recordings. An AIR system should be able to store each of its items efficiently in terms of digital size and also allow for accurate and fast look-up. Users need the ability to rapidly and accurately search for items while working with such systems, which becomes challenging for large collections. Due to the endlessly growing size of information on the internet, the quality of the search results of an AIR system will get only more important with time.

In an AIR system, items can be retrieved in various ways. Traditionally, this is either performed with a user providing a text input that contains its needs or with an example item similar to what is needed. The former is termed Query-by-Text (QbE) and the latter, Query-by-Example (QbE) [4]. An example of a QbT could be a user typing "Fire Crackle" to the search engine while an example of a QbE could be providing a sound clip of a fire crackle. Since sound similarity is a type of QbE, throughout this work we focus on QbE.

Query-by-Example

QbE is finding the most similar items in a collection to a query item. In order to perform such a search, a QbE system stores a numeric representation for each of its items and a mathematical function to rank the similarities between these representations. Representing audio clips has traditionally been done with content-based audio analysis and semantic audio analysis which results in a fixed-length vector of real numbers [5]. Traditionally, these vectors contain features that are extracted from the audio clip such as High-Frequency content, Loudness, and Rhythmic Complexity. Figure 2 demonstrates this process.

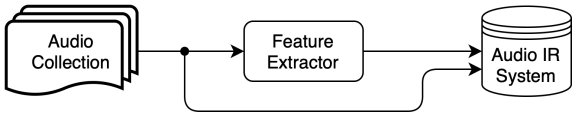

Figure 2: Item and representation storage at an AIR system.

In Euclidean space, there is a variety of functions that input two vectors and output a real number, i.e. a score. Such a function can be used as a similarity measure between its input vectors and hence the sounds that they represent. Here we define the functions that are relevant.

Assume we have a positive integer $D \in \mathbb{Z}^+$ and the corresponding Euclidean Space $\mathbb{R}^D$. Then $\forall\ \mathbf{u}, \mathbf{v} \in \mathbb{R}^D$, Euclidean distance, inner product, and cosine similarity functions are defined in Equations 1.1, 1.2, 1.3 respectively.

$$\|\mathbf{u} - \mathbf{v}\|_2 := \sqrt{\sum_{n=1}^{D} (u_n - v_n)^2} \tag{1.1}$$

$$\mathbf{u} \cdot \mathbf{v} := \sum_{n=1}^{D} u_n v_n \tag{1.2}$$

$$\cos(\theta) := \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2} \tag{1.3}$$

Upon inspecting Equations 1.2 and 1.3, we see that the cosine similarity of two vectors is equal to the inner product of its normalized inputs.

When a QbE is made, a similarity score is calculated between the query item's representation and every other item's representation, using a score function similar to Equation 1.1, 1.2, or 1.3. After similarity scores are calculated, the scores and their corresponding items are ranked from highest to lowest score, and this sequence of ranked items is returned. The similarity search process is demonstrated in Figure 3.

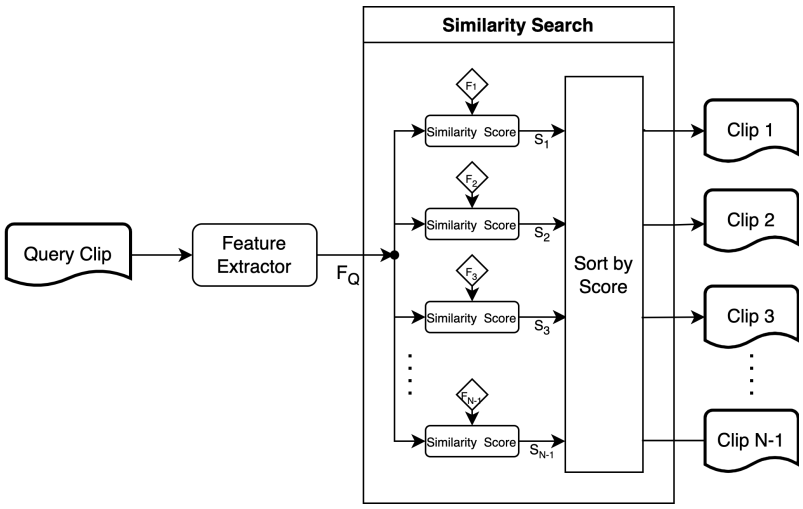

Figure 3: Similarity search in an AIR system.

When the search is based on inner product it is called Maximum Inner Product Search (MIPS), when it is based on cosine similarity it is called Maximum Cosine Similarity Search (MCSS), and when it is based on the Euclidean Distance it is called Nearest Neighbor Search (NNS) [15]. Below we provide their mathematical definitions.

Let Q be the audio collection and $\mathbf{q} \in Q$. Then finding the most similar item in Q to $\mathbf{q}$ can be done by the following equations:

$$\text{MIPS}(\mathbf{q}, Q) := \underset{\mathbf{v} \in Q \setminus q}{\text{argmax}} \ \mathbf{q} \cdot \mathbf{v} \tag{1.4}$$

$$\text{MCSS}(\mathbf{q}, Q) := \underset{\mathbf{v} \in Q \setminus q}{\text{argmax}} \ \frac{\mathbf{q} \cdot \mathbf{v}}{\|\mathbf{q}\|_2 \, \|\mathbf{v}\|_2} \tag{1.5}$$

$$\text{NNS}(\mathbf{q}, Q) := \underset{\mathbf{v} \in Q \setminus q}{\text{argmin}} \ \|\mathbf{q} - \mathbf{v}\|_2 \tag{1.6}$$

By inspecting Equations 1.4, 1.5, and 1.6 we can see that they become equivalent if

$\forall\ \mathbf{v} \in Q$, there exist a positive real number $C \in \mathbb{R}^+$ such that $\|\mathbf{v}\|_2 = C$.

Freesound

Freesound is essentially an AIR system that contains a large collection of audio with a broad range of content. Users can search through the collection by making a QbE search which is called *similar sounds*. It is based on representing the audio clips with a collection of acoustic features. The details of this similarity function are described in Section 3.2.1.

Our experience so far with the sound similarity function has been that it is able to recommend acoustically similar sounds to query clips, especially while working with isolated sounds of musical instruments. It can accurately recommend sounds that were uploaded in the same *sample pack*. Moreover, it can find acoustic similarities between sound sources that have no semantic relation to each other: during our experiments, we were recommended a vinyl crackle sound to a query of fire crackle sound, which sounded *similar* to us. Although this was one successful example out of many unsuccessful attempts, it implies potential applications of the current system.

1.3 Scope and objectives

The main objective of this work is to improve Freesound's sound similarity system semantically by utilizing the recent progress made in deep learning. To this end, we choose a variety of general-purpose deep learning models for audio and explore their semantic capabilities for sound similarity.

In total, we assessed the performance of eight deep learning models. However, the set of models that we could have experimented with was much higher. In this inaugural work, we establish the role of general-purpose deep embeddings for sound similarity and leave a larger comparison for the future. Namely, we focus on audio classification and contrastive multimodal representation models and leave the evaluation of Transformer-based architectures such as the Audio Spectrogram Transformer and

the Contrastive Masked-Autoencoder as future work.

Since our goal is to create a better sound similarity function for Freesound users, we need their input to evaluate the AIR systems that we build. However, before asking the community for feedback, we use objective evaluations supported by our subjective judgment. We leave the large-scale user experiment for future work.

1.4 Summary of the contributions

This work contributes to the field of machine listening, with a focus on the field of semantic sound similarity with machine learning. The main contributions of the thesis can be summarized as follows:

1. A GitHub repository that contains all the computational material related to the project [4].

2. A formal discussion on acoustic, perceptual, and semantic properties of sounds.

3. A critical investigation of the FSD50K evaluation set. Specifically, we trace FSD50K sound class labels back to the Audioset ontology. In the process, we document various classes with semantic issues. We provide a list of the problematic sound classes and audio examples.

4. Rigorous mathematical definitions of objective evaluation metrics for recommendation systems. Although we borrow these metrics from the literature, we improve their mathematical foundations to ensure appropriate calculations.

5. An objective performance evaluation method that takes sound class hierarchies into account. To the best of our knowledge, we are the first to go beyond simple averaging and provide sound family-based averages.

6. A thorough study of the design choices of similarity systems. Specifically, we explore numerous parameters for processing deep embeddings and fundamental similarity search algorithms.

[4] https://github.com/raraz15/freesound-sound_similarity

7. Objective and subjective evaluations of Freesound and three deep learning-based sound similarity systems' performances.

8. An evaluation of supervised and unsupervised deep learning methods regarding their suitability for semantic similarity.

1.5 Structure of the thesis

Chapter 2 provides the scientific background of our methodology. We make a thorough analysis of the FSD50K dataset and its sound classes. Then, we define the relevant objective evaluation metrics. Finally, we provide a review of the state-of-the-art approaches for sound similarity and deep learning methods.

Chapter 3 states our research question and the hypotheses that we formed. We start by describing our methodology for testing the semantic capabilities of sound similarity systems. Then, we describe how we replicate Freesound's sound similarity function and build sound similarity systems with deep learning models.

Chapter 4 presents the results of our experiments. It contains three sections, where each section reports the results related to its corresponding hypothesis. Here, we provide the results regarding our research question and other noteworthy results.

Chapter 5 discusses the results of our experiments in terms of our research question. Each hypothesis is mapped to its results and their validity is tested.

Chapter 6 concludes this thesis. We reiterate our initial goals and summarize how we answered our research question.

Chapter 2

Scientific Background

This chapter provides the scientific knowledge required to understand the methodology employed throughout this thesis. In Section 2.1 we discuss the evaluation dataset used during our experiments. In Section 2.2 we discuss the relevant objective evaluation metrics for evaluating the performances of the built systems. In Section 2.3, we provide a review of the State-of-the-art methods regarding sound similarity systems and deep learning approaches that can be used for sound similarity systems.

2.1 Evaluation dataset

We use the FSD50K dataset to compare the performances of the QbE systems that we build [16]. It is a large collection of sounds taken from Freesound, covering a broad range of sound classes. The audio clips are human-annotated with sound class labels using a subset of the Audioset ontology [17]. Therefore, details of the class hierarchies of Audioset are important for our work.

2.1.1 Audioset ontology

Audioset ontology represents a large set of audio event classes using 632 unique classes coming from seven sound families, which are sound classes themselves. We use underlined notation to denote sound classes: Animal, Natural sounds, Sounds

of things, Music, Source-ambiguous sounds, Human sounds, Channel, environment, and background [17].

The sound classes in Audioset ontology are represented as a hierarchical graph where 474 unique classes are leaf nodes. Therefore, some nodes in the ontology can be a child of different intermediary nodes. For example, Bell is a descendant of the sound family Sounds of things as well as the Music family. Therefore, when we count the appearance of each node once for each appearance, we get 690 total nodes and 527 leaf nodes. We construct the ontology by parsing the Audioset website[1] and distributing it in JSON format alongside our project's repository[2]. We aim to improve the interpretability of this format by creating a new format where sound class hierarchies are represented [3].

2.1.2 FSD50K dataset

The FSD50K dataset contains over 51,000 audio clips which are divided into a development set and an evaluation set. While splitting the audio clips, measures were taken so that all the content of a user went into the same split. Since some of the models that we use during the experiments have been trained on the development set, we use the evaluation set for benchmarking purposes. The evaluation set contains 10,231 audio clips.

With a large-scale effort, the FSD50K dataset audio clips were manually annotated using a subset of the Audioset ontology. The process of collecting the annotations was described in detail in [16]. The evaluation set was labeled *exhaustively*, meaning that considering the label vocabulary, the annotations were complete and correct up to human error. Sound clips are multi-labeled, which means that a clip can be labeled with more than a single label.

[1] http://research.google.com/audioset/

[2] https://github.com/raraz15/freesound-sound_similarity/tree/master/data/ontology

[3] https://github.com/audioset/ontology/blob/master/ontology.json

Sound class hierarchy

The FSD50K audio clips are annotated using 200 labels that are chosen from the Audioset ontology. After analyzing, we saw that all the sound classes have more than 15 examples in the dataset. In order to use these labels for the FSD50K dataset, some decisions were made. We outline the relevant decisions below.

- Channel, environment and background node and its descendants are not included.
- Human sounds, Sounds of things, Source-ambiguous sounds, and Natural sounds nodes' certain descendants are included. However, they are not included as nodes. For example: Squeak is included as a label but its ancestor Source-ambiguous sounds is not included.
- When a node has multiple parents as in the Audioset ontology, it is assumed which parent is chosen. A file that contains these decisions as *propagate to parents* is provided [4]. For example, Squeak can be a descendant of Sounds of things or Source-ambiguous sounds. Following the propagate to parent rule would include it in Source-ambiguous sounds.

Investigating the hierarchy of labels revealed that 22 nodes of the FSD50K dataset have multiple parents in the Audioset ontology. We observed that 8034 of the audio clips are not labeled with these sound classes. We are interested in choosing the right parent and family for these sounds for visualization and performance evaluation purposes. Therefore, we investigate the suggested parents for each of these labels. We found out that out of the 22 nodes, 16 nodes have one suggested parent, four labels have more than one suggested parent and two labels have zero suggested parent. We randomly listen to a minimum of five audio clips per label and discuss our findings here.

[4] https://github.com/xavierfav/Freesound-data-set/blob/master/ontology/ontology_crowd.json

- Clapping can have Hands or Human group actions as a parent and the FSD50K suggested parent is Hands. We observed that audio clips labeled with Hands (e.g. IDs 340356, 119634, and 410868) may include recordings of a crowd. A single pair of hands clapping is both acoustically and semantically different than a crowd clapping together.

- Hiss can be a child of Onomatopoeia, Cat, Snake or Steam. The FSD50K suggested propagation is Onomatopoeia. Audio clips 1942, 89447, and 265581 have all different types of Hiss sounds. Although they are similar acoustically, none of these sounds are semantically similar due to being produced by completely different sound sources.

- Growling can be a child of Dog, Cat, Roaring cats (lions and tigers), and Canidae and dogs and wolves. But there is no suggested propagation. Audio clips 389617, 256646, and 276577 show that a wide range of acoustically similar but semantically dissimilar sounds are included.

- Bicycle bell can be a member of the family Music or Sounds of things and the FSD50K suggests multiple parents such as: Bell, Bicycle, and Alarm. Since all 3 labels exist in the FSD50K, choosing the right parent is difficult. Upon listening we realize that some audio clips contain a bicycle bell recorded on the street while some are recorded in isolation.

Based on these observations we decided not to include the 22 labels in any sound family. We use them as individual labels but do not consider their hierarchical relations. With these changes, we plot and distribute all the sound families together with the repository of the thesis. We provide the Animal sounds and a part of the Music family in Figure 4 and Figure 5.

We evaluate the semantic sound similarity performance of the AIR systems through the use of the sound classes of the FSD50K dataset. Although the semantics of these classes are limited compared to natural language, they are used for initial tests.

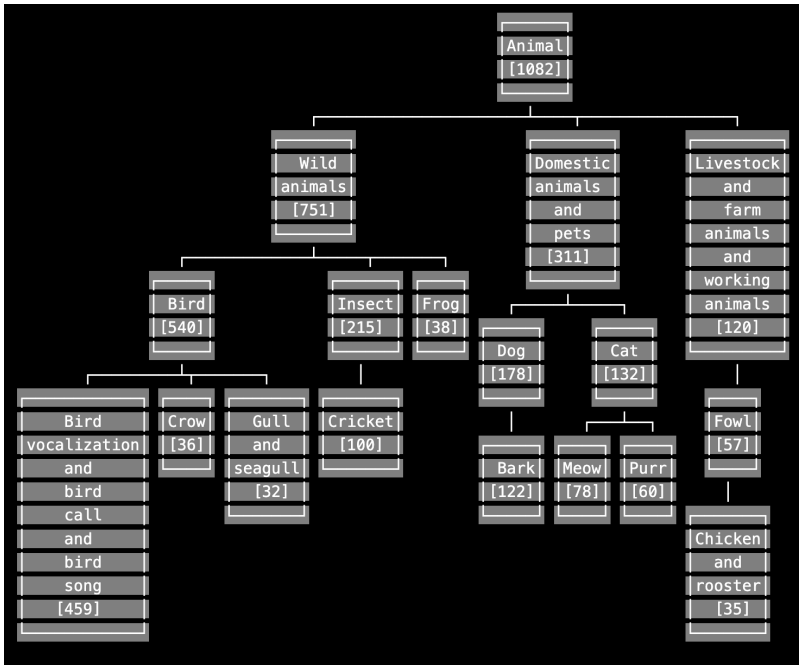

Figure 4: FSD50K animal sounds family. Numbers inside the parentheses indicate the number of occurrences in the evaluation set.

Labeling issues

Throughout the course of this work, we listened to a number of audio clips from the FSD50K evaluation set, which was annotated by online volunteers around the world. Since English was not the first language of everyone who participated in the labeling process, certain sound clips have labeling problems. We document two main problems:

Labels are not always exhaustive. We identified audio clips with missing labels. For example, clip 200809 is labeled as Truck. However, it is the sound of a loud horn with chatter in the background.

There are mislabeled audio clips. We noticed audio clips with false labels. Clip 332516 is labeled as Female speech and woman speaking. However, it is almost exclusively the sound of footsteps in a hallway and the only human voice is a brief whisper.

Our observations about the labeling problems and the issues related to the sound

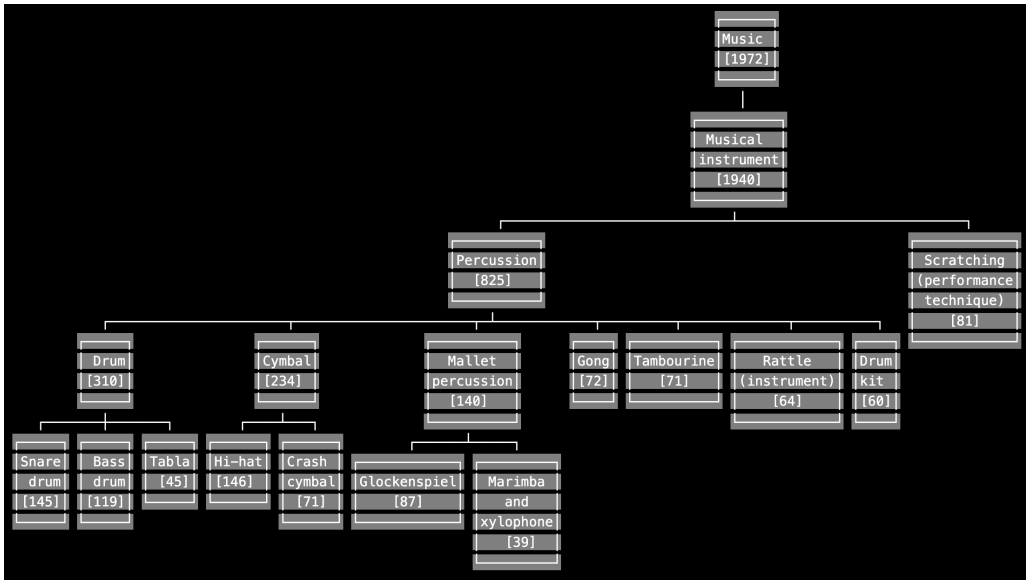

Figure 5: FSD50K evaluation set music sounds family. Numbers inside the parentheses indicate the number of occurrences in the evaluation set.

class hierarchies have potential implications regarding the objective evaluation metrics. In the next section, we define the metrics we use and how we propose to deal with the mentioned shortcomings.

2.2 Objective evaluation metrics

The performance of a QbE system can be evaluated objectively with various metrics that take into account the ordered nature of the results. Such metrics take a single query item and a ranked sequence of items that the system produces to output a score, indicating the performance of the search results. Since a search result is a type of recommendation, we can use recommendation system metrics such as the Average Precision (AP) and the Rank of the 1st Relevant Item (R1). Both metrics require a function evaluating the relevance of two items. We define the sound collection with the following terminology that we use throughout this section.

Let $D, L \in \mathbb{Z}^+$ and assume that $Q = \{\mathbf{q}_1, \mathbf{q}_2, ...\mathbf{q}_D\}$ be a set of audio clips and that we have L distinct sound classes. Then, the audio clips naturally get separated into L overlapping subsets. Let's denote these subsets as $C_j \subseteq Q$, for $j \in \{1, 2, ...L\}$. Since the sounds are multi-labeled, $\exists\ k, l \in \{1, 2, ...L\}$, such that $C_k \cap C_l \neq \emptyset$.

2.2.1 Relevance

We define the relevance of two audio clips based on the relationship between their sound classes. Assume that we are searching for sounds that belong to sound class C_j and we use one such sound to make a QbE. If a recommended sound contains the label C_j we define these two sounds to be relevant.

Then, for $\mathbf{q}_i, \mathbf{q}_j \in Q$, relevance $\sim$ for a sound class $C_t \subseteq Q$ is defined in Equation 2.1. Equipped with the relevance function, we can define metrics that evaluate the quality of a list of query results.

$$\mathbf{q}_i \sim \mathbf{q}_j := \begin{cases} 1 & \text{if } \mathbf{q}_i, \mathbf{q}_j \in C_t \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

2.2.2 R1

When users make a query, they need to see the relevant items as high as possible in the ranking [18]. It may be the case that the first relevant item in the ranking is not in the first position. We define R1 as the ranking of the first relevant item. Depending on the application, a ranking with the lowest R1 can be preferable.

Let Q be the collection of items as above and $\sim$ be the relevance function in Equation 2.1. Also, let $F : Q \longrightarrow Q^{D-1}$ be a function mapping each element $\mathbf{q} \in Q$ to a permutation of $Q \setminus \mathbf{q}$. That is $F(\mathbf{q})_i \in Q$ for $i \in \{1, 2, ..., D-1\}$. Here, F models a recommendation system, $\mathbf{q}$ is a query to the system, and $F(\mathbf{q}) \in Q^{D-1}$ is the ranked search results. Then R1 is defined as in Equation 2.2.

$$\text{R1}(\mathbf{q}, F(\mathbf{q})) = min\ \{i \in \{1, 2, ..., D-1\} \mid \mathbf{q} \sim F(\mathbf{q})_i = 1\} \tag{2.2}$$

2.2.3 Average precision

Average Precision (AP) combines Precision and Recall metrics for retrieval sequences that are ranked [19]. It is a measure of how highly the relevant results are distributed.

Precision is defined as the number of returned documents that are relevant, divided by the number of returned documents. Recall is defined as the number of returned documents that are relevant, divided by the number of total relevant documents. The computation of AP involves Precision@k, which we define below.

For a query $\mathbf{q} \in Q$ and retrieval sequence $F(\mathbf{q})$, for each position $k \in \{1, 2, ..., D-1\}$ in the sequence, Precision@k is defined as:

$$\text{Precision@k } (\mathbf{q}, F(\mathbf{q})_k) := \frac{1}{k} \sum_{i=1}^{k} (\mathbf{q} \sim F(\mathbf{q})_i) \tag{2.3}$$

The AP for $\mathbf{q}$ and $F(\mathbf{q})$ is then defined as:

$$\text{AP}(\mathbf{q}, F(\mathbf{q})) := \frac{1}{R(\mathbf{q})} \sum_{k=1}^{D-1} (\mathbf{q} \sim F(\mathbf{q})_k) \cdot \text{Precision@k } (\mathbf{q}, F(\mathbf{q})_k) \tag{2.4}$$

where $R(\mathbf{q})$ is the number of total relevant documents to $\mathbf{q}$ in the collection, i.e.

$$R(\mathbf{q}) = \sum_{\mathbf{u} \in Q \setminus \mathbf{q}} \mathbf{q} \sim \mathbf{u} \tag{2.5}$$

We would like to emphasize that $R(\mathbf{q})$ is the total number of relevant items in the collection, *not* in the first N positions. Therefore AP is always in the range $[0, 1]$.

When a QbE is made, the system returns all the items in the collection except the query item, ranked from most similar to least similar. However, in a large collection of items, the probability of the user exploring lower rankings is low [18]. Therefore a variant of AP is used to focus on the important part of the results by cutting off the sequence of recommendations after an arbitrary position $N \in \{1, 2, ..., D-1\}$. It is called Average Precision at N (AP@N). The value of N depends on the application. We also update the divisor for taking N into account. Equation 2.6 shows the AP@N calculation.

$$\text{AP@N} := \frac{1}{min\{R(\mathbf{q}), N\}} \sum_{k=1}^{N} (\mathbf{q} \sim F(\mathbf{q})_k) \cdot \text{Precision@k } (\mathbf{q}, F(\mathbf{q})_k) \tag{2.6}$$

2.2.4 Class metrics

We can evaluate the performance of individual query results using AP@N or R1 metrics. However, we are more interested in overall evaluations than evaluating a particular query's results. Therefore, we use the hierarchical relationships between sounds such as sound classes and sound families to compute summarized metrics.

For a sound class, we find the set of audio clips that belong to that class and use this set's elements as queries. We then evaluate each query's results with AP@N or R1 metrics. Finally, we take the arithmetic mean of the metrics. When the arithmetic mean is taken over the AP@N values, it is called the Mean Average Precision@N (mAP@N); and when it is taken over the R1 values, it is called the Mean Rank of the 1st Relevant Item (mR1). Since we calculate these metrics for particular classes, say class X, we call them mAP@N of class X or mR1 of class X. For example, we can calculate mAP@N over the set of Drum kit sounds by using each such sound as a query, calculating the AP@N metric for each query's results, and averaging the set of AP@N values.

Assume C_j is the set of sounds for a particular label. Then, mR1 and mAP@N for that label becomes:

$$\text{mR1}(C_j, F) := \frac{1}{|C_j|} \sum_{\mathbf{q} \in C_j} \text{R1}(\mathbf{q}, F(\mathbf{q})) \tag{2.7}$$

$$\text{mAP@N}(C_j, F) := \frac{1}{|C_j|} \sum_{\mathbf{q} \in C_j} \text{AP@N}(\mathbf{q}, F(\mathbf{q})) \tag{2.8}$$

We can calculate class metrics to evaluate the performance of a QbE system for particular classes. Since we are using multiple elements as queries, a class metric is less susceptible to outliers or labeling errors, which were discussed in Section 2.1.2. Equation 2.7 and Equation 2.8 use each element only once, hence they are referred to as micro-averaged metrics.

We would like to close this section by emphasizing that the mAP metric in Information Retrieval Systems is not related to the metric with the same name in

Classification Systems, where elements are not in ranking relationships.

2.3 State-of-the-Art

Recall Figure 3. A similarity search system has 2 components: an audio representation system and a similarity search system. Although there are different approaches to the similarity search system, the main component is the representation system. Existing approaches differ in the sound properties that are modeled which dictates how the representations are extracted.

Representing audio clips with low-dimensional vectors has been crucial for audio applications. Traditionally, audio representations were obtained by extracting handcrafted features from audio clips and performing various transformations. We provide a brief review of the traditional systems in Section 2.3.1. With the rise of Neural Networks, the traditional approach was replaced with obtaining representations from NNs, which is called Representation Learning [20]. In Section 2.3.2 we report the QbE systems that were built with representation learning. Since our goal is to identify other representation learning methods, in Section 2.3.3 we report a variety of models that were trained without QbE purposes.

2.3.1 Traditional QbE systems

Traditionally, audio representations in sound similarity systems were created by carefully selecting a set of features and combining them in meaningful ways [4], [5], [6]. Mostly, the chosen features were modeling acoustic properties of sound. Based on the use case, some approaches also considered perceptual or semantic properties as well. Below we mention a few such systems.

Barrington et al. represent audio clips with a semantic feature vector with the Gaussian Mixture Model [4]. They learn a mapping from MFCC features to words using a dataset of captioned audio clips. For computing similarity, they use the Kullback–Leibler divergence. Wu et al. make a thorough comparison between various acoustic features and measure their perceptual similarity by asking participants for

their opinions using the Mean Opinion Score (MOS) method [6]. Mechtley et al. combine acoustic, semantic, and social similarity [5]. They extract loudness, STFT magnitude spectrum, MFCCs features to model the acoustic content and use the WordNet lexical database to measure semantic similarity.

2.3.2 QbE systems that use deep learning methods

Jansen et al. create multiple audio representations using supervised metric learning based on sound classes and unsupervised contrastive learning based on temporal proximity constraint, respectively [21]. They compare the semantic properties of these representations in a QbE setting and report that the supervised representations perform better. However, we find their evaluation method that excludes negative, negative pairs from the ranked sequences lacking.

Sert and Basbug combine acoustic and semantic similarity by creating representations using acoustic features and WordNet [9]. They process the features using convolutional and recurrent neural networks in a cascade. For acoustic similarity, they use Euclidean distance as in the acoustic space and the path similarity method in the semantic space. Manocha et al. trained a Siamese network to obtain audio representations. They test the semantic properties of the representations in a QbE setting on ESC-50 and USK8K datasets but they report low mAP values [8]. Fan et al. consider semantic and acoustic properties and use a siamese network to obtain audio representations [10]. They use a pairwise distance metric for measuring the similarity of multi-label sounds.

2.3.3 Representation learning for general purposes

Representation learning refers to learning useful data representations from the data itself [20]. Often, the usefulness of learned representations results from their lower dimension than the original data. Representation learning can be achieved with supervised learning or unsupervised learning methods. Obtaining the actual representation is termed *taking embeddings*. Below we describe State-of-the-art models for representation learning and their embedding taking process.

Supervised learning methods

Hershey et al. use a variety of Convolutional Neural Networks (CNN) to classify the audio clips of a large collection of YouTube videos with their video titles [11]. They find out that the ResNet-50 model works the best, but since a VGG variation has a comparable performance with fewer parameters and less training, they use the VGG variant, hence the name, VGGish. YAMNet model was released by TensorFlow without an accompanying paper [22]. It is trained to classify the audio clips of the Audioset Corpus using the classes of the Audioset ontology. It has 1/20th the weight of VGGish.

Kong et al. pretrains a Neural Network (NN) that uses both the waveform and the log-mel spectrogram on Audioset, which they call Pretrained Audio Neural Networks (PANNs) [23]. After the pretraining, they used transfer learning to six audio pattern recognition tasks and achieved state-of-the-art performance at several of them at the time. Fonseca et al. implement signal-processing-based architectural changes to the VGGish model to improve the shift-invariance property of the NN and evaluate the effects on FSD50K [13]. Gong et al. also use the AudioSet, however, underline the importance of the training strategy [24]. They train an EfficientNet model using a variety of model-agnostic training methods such as: pretraining on image datasets, sampling strategies, data augmentation techniques, and model aggregation. On both AudioSet and the FSD50K datasets they achieve the state-of-the-art performance [16]. Verma and Berger use a transformer-based architecture to classify audio clips but do not utilize pretraining as in other domains to surpass convolutional architectures [25].

For models that were trained with supervised learning, the last layer is usually a classifier layer and the rest of the network provides a representation of this classifier [20]. It is common to take the outputs of this penultimate layer as a representation of the audio. These models make frame-level predictions which are then aggregated to a single clip-level prediction [11].

When supervised models are used, semantic properties of sounds are integrated

through the use of sound class labels. As the weights of the network are updated together, while the classification head is moving towards classifying its inputs with correct labels, the rest of the network is moving towards creating a representation that can capture semantics through labels.

2.3.4 Unsupervised learning methods

In unsupervised representation learning, the task is to learn representations without supervision [20]. Therefore the outputs are directly taken as embeddings. Recently, a form of unsupervised learning became popular: self-supervised learning, where the supervision is the data itself. Data supervises itself based on our assumptions about the data or by looking at itself from different perspectives, i.e. modalities. In cases where more than one modality is used, networks learn useful audio representations by combining information from each modality. Therefore, when language is used with audio, audio representations obtain semantic properties [26]. Models that align the representation of multiple modalities deal with variable-length audio internally so that the resulting embeddings do not have the time dimension.

Saeed et al. train a unimodal representation model on audio [27]. They assume that the temporal proximity of audio events imply a shared source to train a contrastive representation learning model. Arandjelovic et al. train a bimodal representation network using the audio and images from videos [28]. They train the network with the audio-visual correspondence task and call the model Look-Listen-Learn (L3). Cramer et al. integrate audio domain design choices to L3 and make the model public and call it OpenL3 [29]. OpenL3 has multiple variations that are trained on the music and environmental sounds subsets of Audioset corpus.

Favory et al. use audio tags as a second modality to audio and jointly train an audio encoding network with a tag encoding network on a sound collection obtained from Freesound. [30]. Elizalde et al. use natural language and audio to perform contrastive language-audio pretraining, hence the name CLAP [26]. The modal is trained on audio-text pairs selected from a combination of FSD50K, ClothoV2, AudioCaps, and MACS datasets.

Recent approaches combined 3 modalities: text, image, and audio. Wu et al. distill information from a pretrained CLIP text, image bimodal model to teach an audio network, AudioCLIP [31]. The distillation takes place on AudioSet and CLIP was trained on the CLIP dataset. Guzhov et al. use a multi-stage training process that includes a pre-trained CLIP model [32]. They pre-train an ESResNeXt network on AudioSet and then train the CLIP model together with ESResNeXt on AudioSet. Girdhar et al. align 5 modalities individually to the image, hence the name ImageBind [33]. The model has strong emergent properties. Even though it is not trained with text and audio pairs, it can perform zero-shot classification between these two modalities.

Chapter 3

Methodology

To answer our research question of whether Freesound's sound similarity function can be improved in terms of semantics, we test the three following hypotheses.

$\mathcal{H}_1$**:** Freesound's sound similarity function does not capture semantic properties of sound sufficiently.

$\mathcal{H}_2$**:** General purpose deep embeddings can be used to create audio representations that capture semantic sound properties.

$\mathcal{H}_3$**:** For semantic sound similarity, audio representations obtained from deep embeddings can work better than Freesound's audio representation.

Section 3.1 discusses the objective and subjective evaluation methods we use for assessing system performances. In Section 3.2 we replicate Freesound's sound similarity system and evaluate it in terms of semantic sound similarity. Then, in Section 3.3 we integrate a variety of deep learning models into sound similarity systems and choose the best deep learning model for semantic sound similarity. Finally, in Section 3.4 we describe the comparison between Freesound's sound similarity system and the best system that uses a deep learning model.

3.1 Performance evaluation

To evaluate the performances of QbE systems and to compare the performances of multiple systems we perform objective and subjective evaluations.

3.1.1 Objective evaluation

The basis of our performance evaluations is based on the mAP@N metric for individual classes, which was defined in Equation 2.8. However, this fine-grain analysis can be hard to interpret. When we consider that there are 200 sound classes in our evaluation set (See Section 2.1.2), it can be hard to use this metric for comparing multiple models' performances. Therefore, we calculate various summarized metrics that are based on mAP@N and mR1.

Total composite metrics

To summarize the performance of a system, for both mR1 and mAP@N metrics, we average all classes' metrics to get a single, composite metric representing the performance over the entire dataset. We call these metrics as the total composite mAP@N and mR1.

Recall that we have L labels in the collection and $Q = \bigcup_{k=1}^{L} C_i$.

$$\text{mR1}_{total}\{\{C_1, C_2, ...C_L\}, F\} := \frac{1}{L}\sum_{k=1}^{L}\text{mR1}(C_k, F) \tag{3.1}$$

$$\text{mAP@N}_{total}\{\{C_1, C_2, ...C_L\}, F\} := \frac{1}{L}\sum_{k=1}^{L}\text{mAP@N}(C_k, F) \tag{3.2}$$

We use the metrics in Equation 3.1 and Equation 3.2 as the basis for comparing the performances between models. To mark that they are composite metrics we use the *total* subscript.

Family composite metrics

Considering the sound hierarchies that were described in Section 2.1, we group the sound classes into sound families such as Animal or Music and excluded the problematic sound classes as described in Section 2.1.2. We then compute a composite metric that uses the mAP@N metrics of multiple sound classes. For example, for the Animal family, we average the mAP@N values of its member classes. We call the resulting metric: the animal family's composite mAP@N. We do not provide the family-composite mR1 values to not clutter the thesis, however, during our experiments we use it for evaluation.

Assume that a sound family T has K member classes with corresponding sets $C_{T_1}, C_{T_2}, ...C_{T_K} \in Q$, i.e $T = \{C_{T_1}, C_{T_2}, ...C_{T_K}\}$. Equation 3.2 defines the family-composite mAP@N metrics.

$$\text{mAP@N}_{family}\{T, F\} := \frac{1}{K}\sum_{k=1}^{K}\text{mAP@N}(C_{T_k}, F) \tag{3.3}$$

Due to the multi-label nature of the FSD50K dataset, an audio clip may be a member of multiple sound classes. Therefore, the metrics in Equations 3.1, 3.2, and 3.3 are referred to as macro-averaged in the literature.

It's worth mentioning the influence of label accuracy on our computed metrics. Any mislabeling in a pair of sounds can influence their relevance value. For example, a false irrelevance can result from a missing label, or an incorrect relevance can result from a wrong label. While instances of mislabeling do exist in the FSD50K dataset, they're relatively low in comparison to accurate annotations (a detailed observation was shared in Section 2.1.2). Despite the potential impact on metrics for certain labels, the effect is somewhat avoided by averaging metrics across numerous examples.

3.1.2 Subjective evaluation

We develop a user interface that simulates the sound similarity function. It allows us to choose a sound class and sample a random sound from that class. The sampled sound and the similar sound recommendations of each AIR system are displayed side-by-side. Figure 6 demonstrates the interface.

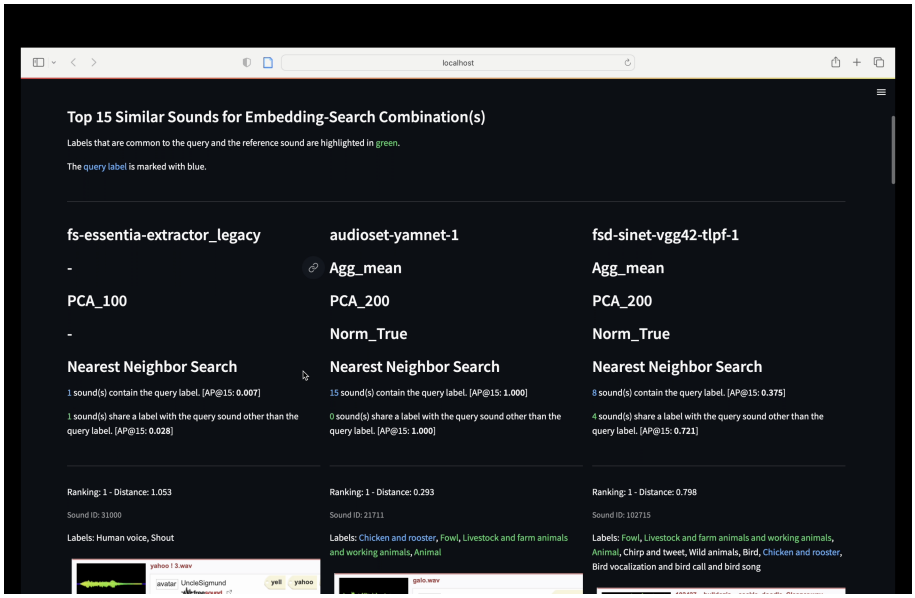

Figure 6: Subjective evaluation interface.

The interface can display similar sound recommendations of up to 4 QbE systems together. Each system's similar sound recommendations are accompanied by the AP@N metric of its results. We display sound class information with color coding. Each label is displayed in white, and for a recommended sound, its labels that are shared with the query sound are marked green. In the case that the query label exists in a recommended sound, it is marked with blue. Moreover, for each query sound; the Freesound ID, ranking of the sound in the list, and the similarity score are also displayed.

3.2 Evaluating Freesound's sound similarity

One of our main goals is to assess the semantic sound similarity capabilities of Freesound. To accomplish this, we first recreate its sound similarity system using the

FSD50K evaluation set as an audio collection. Then, we evaluate the performance of this system that mimics Freesound's sound similarity system through a range of objective performance metrics.

3.2.1 Replicating Freesound's sound similarity

Freesound extracts a mixture of low-level and high-level features from audio clips [1] [1]. Also, the Freesound API can be used to request computed features for sounds that are uploaded to Freesound. The full list of features can be accessed from [35]. In total, 37 of these features are hand-picked to represent the audio clip. Some of these features are single-band, such as Dissonance, Pitch, and Spectral Crest; while others are multi-band, MFCC, Barkbands, and Spectral Contrast. We note here that some features such as Spectral Crest and Spectral Contrast are non-perceptual while MFCC, Dissonance, and Pitch are perceptual. Therefore, the similar sound function of Freesound models only the acoustic content of sound, and the semantic content is not directly taken into account.

For each feature, statistics such as the mean, mean of the derivative, mean of the second order derivative; variance, derivative of the variance, and the second order derivative are computed over time. Therefore, for each feature and its statistic, we get a single number for the entire audio clip, i.e. the features are already aggregated over time. The single-band features are represented as a vector of 6 dimensions containing these statistics. Multi-band is split into multiple single-band features. In summary, a mixture of 37 single-band and multi-band features are split into 141 features, each having 6 dimensions.

The 141 features are min-max scaled to $[0, -1]$ independently by finding the minimum and maximum value of that feature between all the audio clips. The normalized features are concatenated, and the resulting representation has 846 dimensions. In order to reduce the dimension of the audio representations, the current implementation applies Principal Component Analysis (PCA) to reduce the final dimension to

[1] The code that can extract these features is not open to the public, but it uses an old version of Essentia [34]

100 components. We should note here that, since Freesound contains a vast amount of audio files that is growing by the minute and fitting the PCA algorithm is costly, authors only use the first 10,000 audio clips' features to fit the PCA algorithm. Since the FSD50K evaluation set contains 10,231 audio clips, we also use all of the audio clips to fit the PCA algorithm in our experiments.

The similarity score is calculated using the Euclidean distance and the similarity search is done by NNS as described in Equation 1.6. The choices of these functions were not due to experimentation, but due to their availability in the Solr software that performs the search [36].

3.2.2 Analyzing Freesound's QbE performance

We evaluate Freesound's sound similarity performance on multiple levels using the metrics that were described in Section 3.1.1. We first calculate the mAP@15 and mAP@150 metrics of each sound class. As Freesound displays 15 sounds on each page, these metrics reflect the performance on the 1st and 10th pages, respectively. We also calculate the mAP@15_{family} and mAP@150_{family} metrics for each sound family to understand the performance of different sound families.

To facilitate performance comparison between models, we use the mAP@15_{total}, mAP@150_{total}, and mR1_{total} metrics. Using both the mAP@15_{total} and mAP@150_{total} we can evaluate the consistency of the search results over a larger range. Using the mR1_{total} additionally opens a new perspective for evaluation.

3.3 Sound similarity with deep embeddings

A key objective of our research is to see if general-purpose deep embeddings can be used as representations in a semantic sound similarity system. Section 3.3.1 describes the integration of deep learning models into sound similarity systems. Section 3.3.2 describes how the best deep learning model for semantic similarity is identified.

3.3.1 Integrating deep embeddings in QbE systems

The following sections describe the necessary steps for integrating deep learning models in QbE systems.

Model selection

In Section 2.3 we reviewed a variety of deep learning models tailored for both general purpose and specific tasks for sound similarity that are suitable for taking embeddings. However, even though the models described in Section 2.3.2 were trained specifically for sound similarity, they are not publicly available. Therefore, we choose a variety of supervised and unsupervised models for taking embeddings.

From the supervised learning models reviewed in Section 2.3.3, we choose YAMNet, VGGish, and FSD-SINet [37], [11], [13]. We choose YAMNet and VGGish as they are fundamental models in audio classification and FSD-SINet because it was specifically trained on the FSD50K development set. From the unsupervised models reviewed in Section 2.3.4 we choose OpenL3 to test the semantic capabilities of audio-visual models; AudioCLIP and Wav2CLIP to test the semantic capabilities of trimodal models; CLAP to test the semantic capabilities of language-audio models; ImageBind to test the semantic capabilities of its unique 6 modalities, [29], [26], [33], [32], [31].

Embedding processing

Unsupervised models such as CLAP and ImageBind have the advantage of representing variable-length audio clips with a single embedding, i.e. clip-level embeddings. This is a desirable property due to storage space and computation time. Conversely, supervised models such as YAMNet and FSD-SINet produce embeddings at the frame level. Therefore, their embeddings need to be aggregated over time to obtain a single, clip-level embedding. The prevalent approach for frame aggregation is using averaging [11], [29]. However, this approach may smear important sound characteristics in longer audio clips due to the smoothing effect of averaging. In contrast, distributional statistics such as the median of a distribution is more robust

to the number of elements and the maximum of a distribution will never be lower than its mean. In order to make sound sources more apparent, we also experiment with median and max aggregation methods.

Following the aggregation step, we experiment with PCA. We start with preserving the original embedding size by not applying PCA. Then we create an alternative representation by applying PCA to keep 100 principal components to have an even comparison with Freesound's 100-dimensional representation. Finally, we create a third representation using PCA, setting the number of retained components to either 200 or a value less than 100, depending on the original embedding dimensions of each NN.

Finally, we experiment with normalizing the vectors to have unit norm or keeping the vectors as they are. Following the aggregation, PCA, and normalization processes, the resulting vectors are used as audio representations.

Similarity search

Once we obtain a representation for each audio clip, we continue exploring the similarity search algorithms that were described in Section 1.2.2. Our motivations for this experiment are: firstly, to find out whether there is a benefit in storing normalized inputs, which would save computation time during the similarity search; and secondly, to discover the potential benefits of particular algorithms.

Each pretrained NN, together with an embedding processing and a similarity search function creates a unique QbE system. For example, a unique QbE system is created by using the YAMNet model to take embeddings, applying mean aggregation, reducing dimensions via PCA to 100 components, normalizing vectors, and finally utilizing MIPS to perform the similarity search.

In Section 1.2.2, we established that MIPS, MCSS, and NNS become equivalent when the vectors are normalized. With this information, we enumerate the various Query-by-Example (QbE) systems explored. For each supervised model we experiment with three types of aggregation, three types of PCA values, and three types

of search algorithms resulting in 27 unique systems. On the other hand, unsupervised models output clip-level representations, thus we experiment with nine distinct QbE systems. Some unsupervised models such as AudioCLIP produce normalized embeddings. Hence, for such models, we effectively experiment with seven systems.

3.3.2 Finding the best embeddings

Of the eight selected models; YAMNet, VGGish, Wav2CLIP, and ImageBind each have a single variation, while the remaining models have multiple variations. For example, FSD-SINet has four variations, including VGG41 tlpf_aps and VGG42 tlpf. Our goal is to find the best QbE system possible for each model, which includes finding the best variation for the models with multiple variations. Therefore, initially, we treat each variation as a separate model and search for the best QbE system possible for each.

As described in Section 3.3.1, we create up to 27 QbE systems per model, depending on the used NN. Between these systems, we identify the best one based on the mAP@15$_{total}$ metric and discard the systems with lower performance. This approach drastically reduces the search space. Instead of performing a detailed analysis to compare multiple QbE systems per NN, we do a broad comparison based on the mAP@15$_{total}$ value.

However, as mentioned above, some models have multiple variations. Therefore, we group the models based on their variations and select the best variation from each group using the mAP@15$_{total}$ metric. For example, we compare the four QbE systems that use different FSD-SINet variations and choose the best variation. Ultimately, we obtain eight distinct QbE systems, each utilizing a unique pretrained NN.

To find the best deep learning model for semantic sound similarity, we carry out a comparative evaluation of these eight fine-tuned QbE systems. The metrics employed for this final comparison are the mAP@15$_{total}$, mAP@150$_{total}$, and mR1$_{total}$, building upon the approach used for cross-model comparisons. Based on these metrics, we identify the best NN-based QbE system. We then subject it to a thorough

analysis following the methodology outlined for Freesound in Section 3.2.2.

3.4 Feature engineering vs deep embeddings

To answer our research question, we carry out a two-step final assessment. In the first step, we compare the performance of Freesound's QbE system with the three best-performing NN-based QbE systems, which were identified in the previous section. This comparison is based on the mAP@15$_{total}$, mAP@150$_{total}$, and mR1$_{total}$ metrics, consistent with our previous objective evaluations.

In the second step, we perform a thorough subjective evaluation with the web interface introduced in Section 3.1.2. Using the interface, we make multiple sound queries and listen to similar sound recommendations of Freesound together with the three best-performing NN-based systems. Our goals with the subjective evaluation include witnessing similar sound recommendations ourselves, assessing whether semantic similarity is captured sufficiently through sound classes, evaluating the relevance between recommended sounds by listening, and seeing if our listening experience aligns with the objective metrics.

In each stage, we can subjectively evaluate the systems. However, using the interface, listening to a single page of results of a single QbE system takes approximately 5 minutes. When our search space is considered, it is not feasible to evaluate each system in this way. Therefore, during the initial steps of the evaluation, we use only the objective evaluation methods and in the final step, we validate the effectiveness of the objective evaluation methods with our subjective evaluation.

Chapter 4

Results

In this chapter, we provide the results of the experiments that were described in Chapter 3. Section 4.1 reports the performance evaluation of Freesound's sound similarity system. Section 4.2 reports how we choose the best sound similarity system that uses deep embeddings. Section 4.3 presents the performance comparison between Freesound and the best sound similarity system that uses deep embeddings.

4.1 Evaluating Freesound's sound similarity

We replicate the sound similarity system of Freesound, as described in Section 3.2.1. Here we provide this system's objective performance evaluation results. We employed the mAP@15, mAP@150, and mR1 metrics; and using the sound hierarchies, we evaluated the performance across various levels: from individual classes toward sound families, to summary metrics.

Figure 7 demonstrates Freesound's mAP@15 scores for sound classes. We observe that, for each class, the score is bounded above by 0.5, and for the majority of the classes it is below 0.1. We also observe that the top 10 sound classes are mainly musical instruments. Finally, for sound classes such as Tick, Boiling, and Vehicle horn and cor horn and honking the metric is almost 0.

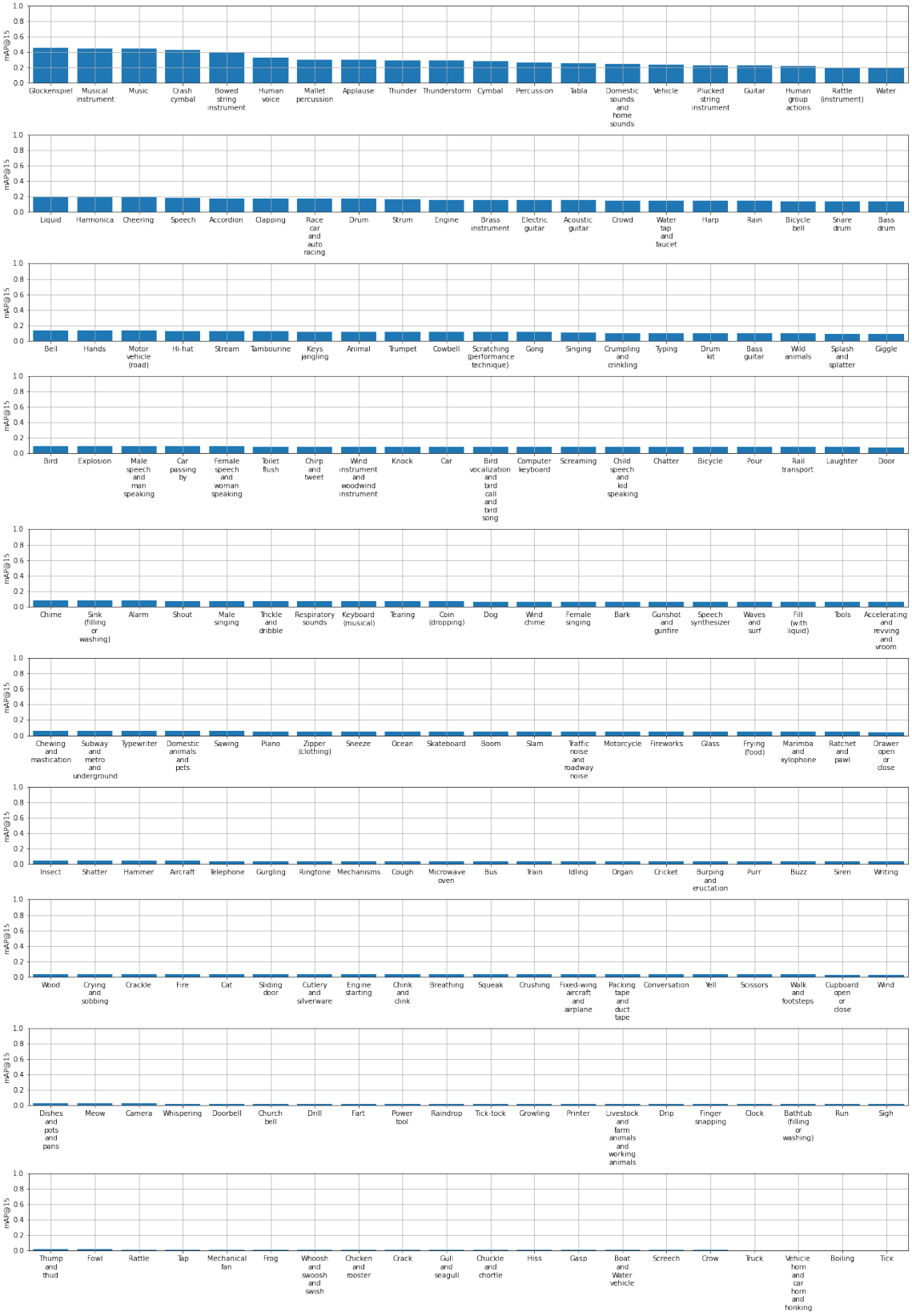

Figure 7: Freesound's semantic sound similarity performance on sound classes by mAP@15.

Table 1 reports the mAP@15$_{total}$, mAP@150$_{total}$, and mR1$_{total}$ metrics for Freesound. The results show that on average for a sound class, the mAP@15 score is 0.09, and mAP@150 is 0.03. Which are both far from the maximum value of 1.0. Moreover, on average for a sound class, the first relevant result appears on ranking 43.2, at the bottom of page 3.

mAP@15$_{total}$ (↑)	mAP@150$_{total}$ (↑)	mR1$_{total}$ (↓)
0.09	0.03	43.2

Table 1: Freesound's sound similarity performance by various total-composite metrics.

Figure 8 shows the mAP@15$_{family}$ and mAP@150$_{family}$ metrics for all sound families for Freesound. We observe that Freesound's sound similarity works best for Music and worst for Animal sounds. The difference in performance between the families is notable.

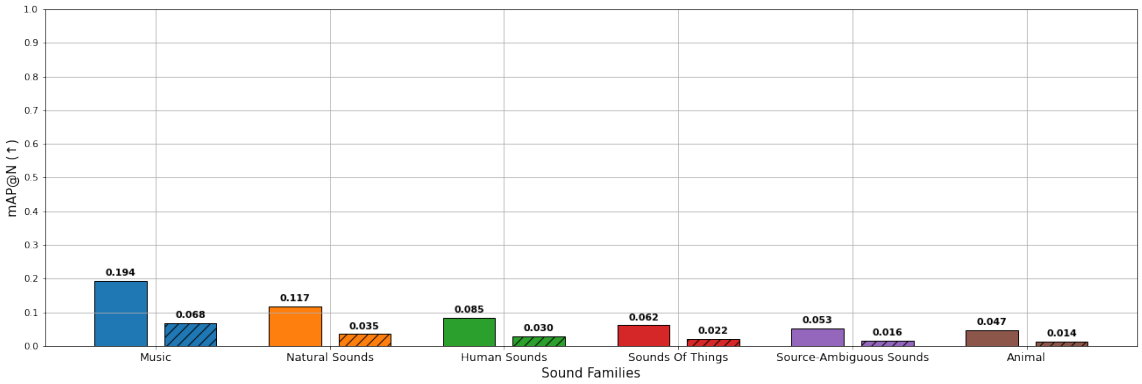

Figure 8: Freesound's semantic sound similarity performance on sound families by mAP@N$_{family}$. Dashed blocks represent N=150 and simple blocks represent N=15.

4.2 Sound similarity with deep embeddings

In this section, we report the results of the experiments that were described in Section 3.3 related to the integration of deep embeddings in sound similarity systems.

4.2.1 Integrating deep embeddings in QbE systems

We select 8 deep learning models and look for the best sound similarity system that can be created for each model. As described in Section 3.3.1, supervised and unsupervised models have different embedding processing steps. Therefore, we provide results from an exemplary model from each model type separately.

Figure 9 demonstrates the objective evaluation of sound similarity models that take embeddings from the VGG42 tlpf variation of FSD-SINet, a supervised model. The x-axis contains embedding processing parameters while the y-axis corresponds to the performance metric values. The color of a block indicates the similarity search algorithm that was used with a given embedding processing system. Therefore, between a group of three blocks that are displayed together, the same embedding processing system was used with three different search algorithms. Hence, each block corresponds to a unique QbE system. This kind of visualization aids in understanding the effectiveness of different combinations of embedding processing steps and similarity search functions. By comparing the metrics across different QbE systems, we can identify the best set of parameters given a deep learning model.

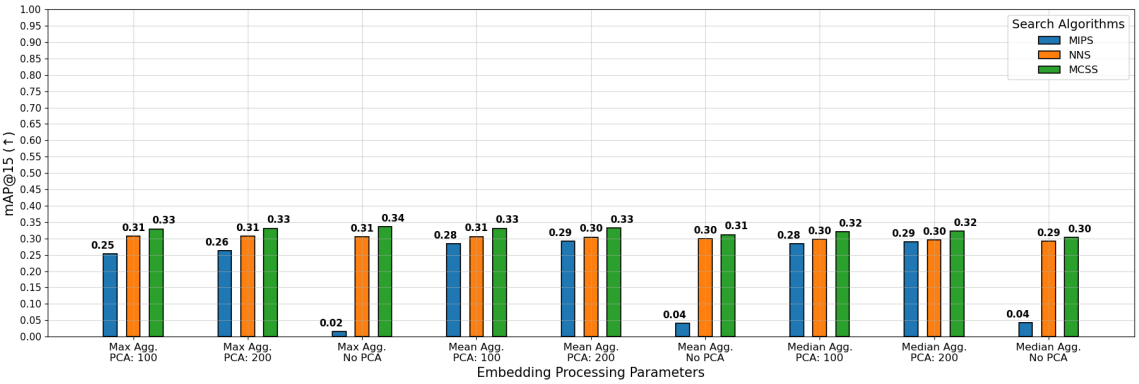

Figure 9: Semantic sound similarity performances of QbE systems that use FSD-SINet VGG42-tlpf by mAP@15$_{total}$.

We make the following observations from Figure 9

- **Effectiveness of MCSS**: For any embedding processing system, between the three search algorithms, the highest score was achieved with MCSS.

- **Limitations of MIPS**: When no dimension reduction via PCA was applied, MIPS scored particularly lower. Additionally, regardless of PCA, when the representations did not have unit length, MIPS scored lower.
- **Highest Metric with Maximum Aggregation**: The highest metric was achieved with maximum aggregation. The difference between the mean aggregation, which achieved the 2nd position, is less than 3%.
- **Metric Sensitivity to Aggregation Type**: The aggregation type did not influence the metric considerably. We posit that this is due to the fact that the models were trained with mean aggregation, which may have influenced their distribution.
- **Impact of PCA**: For mean and median aggregation types, dimension reduction with PCA increased the metric with respect to not applying PCA.
- **Dimensionality and Performance**: Decreasing the dimensions to 100 had no significant decrease in the metric for any aggregation and search algorithm combination.

These insights help us better understand the influence of different processing steps and algorithmic choices on the performance of sound similarity systems, specifically using the FSD-SINet model. They provide valuable guidelines for fine-tuning a system to achieve optimal performance.

For conciseness, we omit the performance plots of other supervised models. However, even though for FSD-SINet VGG42 tlpf, maximum aggregation scored the highest; for VGGish, mean scored the highest; and for YAMNet, maximum scored similarly to mean. This observation indicates that the effectiveness of aggregation methods can be model-dependent, i.e., different pretrained neural networks may respond differently to the same type of aggregation.

Figure 10 demonstrates the objective evaluation of the sound similarity systems that take embeddings from CLAP, an unsupervised model. Its outputs are already aggregated in time, so we experiment with fewer design parameters compared to a

supervised model. Also, by default, its outputs are normalized to have unit length. Therefore, the three rightmost systems had exactly the same metric for different search algorithms (see Section 1.2.2 for more explanation). Finally, we observe that reducing the dimension to 100 principal components has close to no reduction in the metrics for any search algorithm, which may indicate that the representations are sparse.

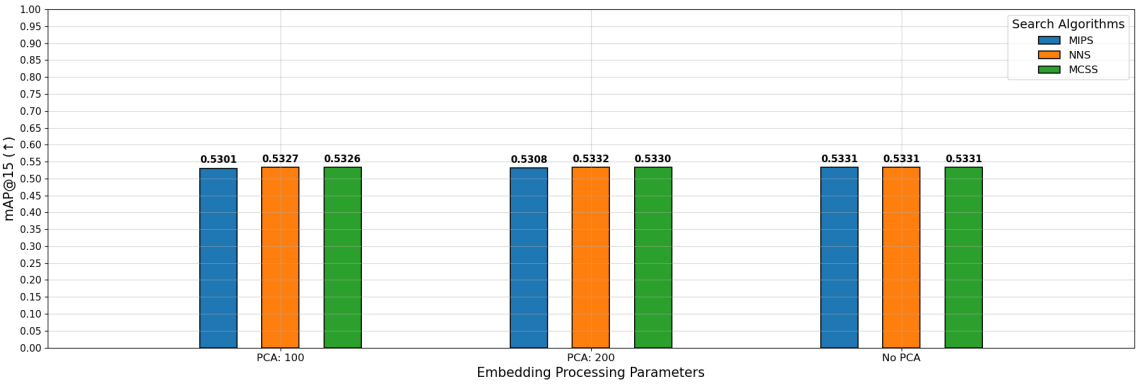

Figure 10: Semantic sound similarity performances of QbE systems that use CLAP by mAP@15$_{total}$.

As described in Section 3.3.2 some models such as the FSD-SINet or OpenL3 have multiple variations. For such models, we repeat the experiments for each variation and choose the best-performing variation to represent that model. In Table 2, we provide the names of the variations that performed the best.

Model	Variation
FSD-SINet	VGG42 tlpf
OpenL3	Mel256-Emb512
CLAP	630k-fusion-best
AudioCLIP	Full Training

Table 2: Best performing variations of multiple variation models.

Results of the experiments for supervised and unsupervised models show that, applying mean aggregation, keeping 100 PCA components, and normalizing the representation results in close-to-the-best performance for each model. Therefore, we

fix these parameters for all the embedding processing systems - except for the mean aggregation, which is fixed for supervised models only. Moreover, since Freesound uses the NNS similarity search algorithm and all three search algorithms that we consider are equivalent for normalized vectors (see Section 1.2.2), we choose NNS for all similarity search systems.

4.2.2 Finding the best embeddings

We plot the objective performances of all sound similarity systems with mAP@15$_{total}$ and mAP@150$_{total}$ metrics in Figure 11 and with mR1$_{total}$ in Figure 12.

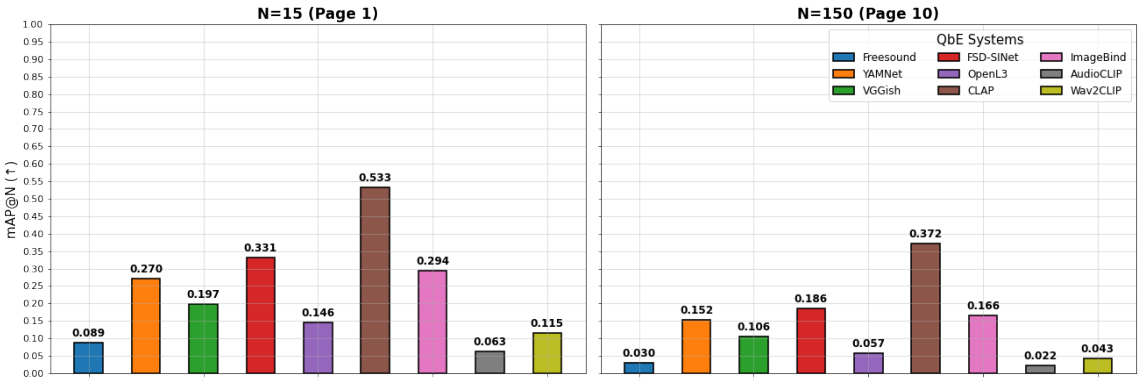

Figure 11: Objective evaluation of all QbE systems in terms of semantic sound similarity by mAP@15$_{total}$ and mAP@150$_{total}$. Each color represents the performance of a unique QbE system.

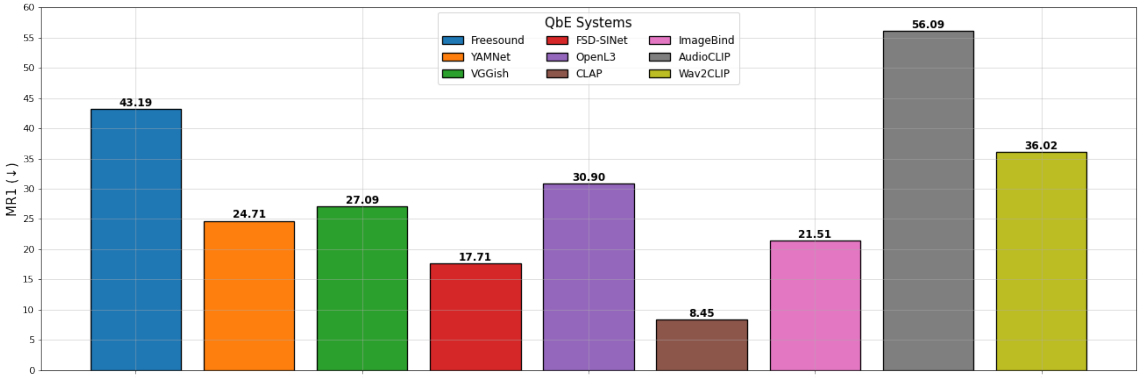

Figure 12: Objective evaluation of all QbE systems in terms of semantic sound similarity by mR1$_{total}$. Each color represents the performance of a unique QbE system.

In Figure 11, CLAP has a higher score than all other models in both mAP@15$_{total}$ and mAP@150$_{total}$ metrics, followed by FSD-SINet and ImageBind. OpenL3, Au-

dioCLIP, and Wav2CLIP scored significantly lower than other models. YAMNet and VGGish score between FSD-SINet and OpenL3. A consistent ranking between models is observed in Figure 12, where CLAP achieves the lowest MR1 score, further verifying its higher performance. Since CLAP performed better in terms of each metric, we further analyzed its performance on individual sound classes and sound families, following Freesound's evaluation method.

Figure 13 demonstrates CLAP's performance for individual sound families with the mAP@15$_{family}$ and mAP@150$_{family}$ metrics. We see that it scores the highest for Music sounds and the lowest for Source-ambiguous sounds. We see that for the majority of the families, the mAP@15$_{family}$ score is higher than 0.5.

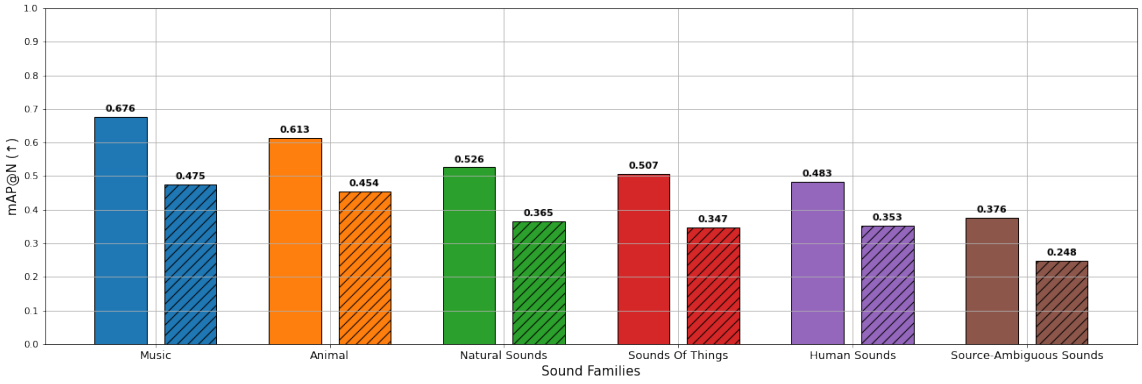

Figure 13: Semantic sound similarity performance of CLAP by mAP@N$_{family}$. Dashed blocks represent $N = 150$ and simple blocks represent $N = 15$.

In Figure 14, CLAP's mAP@15 scores are predominantly above 0.6 across varied sound classes. Exceptionally high scores are noted for classes such as Toilet Flush, Cat, and Burping and Eructation, indicating strong semantic capabilities for these sounds. Conversely, low scores for classes such as Boat and Water Vehicle, Truck, and Tap suggest limitations in the model's performance for these categories, requiring further investigation.

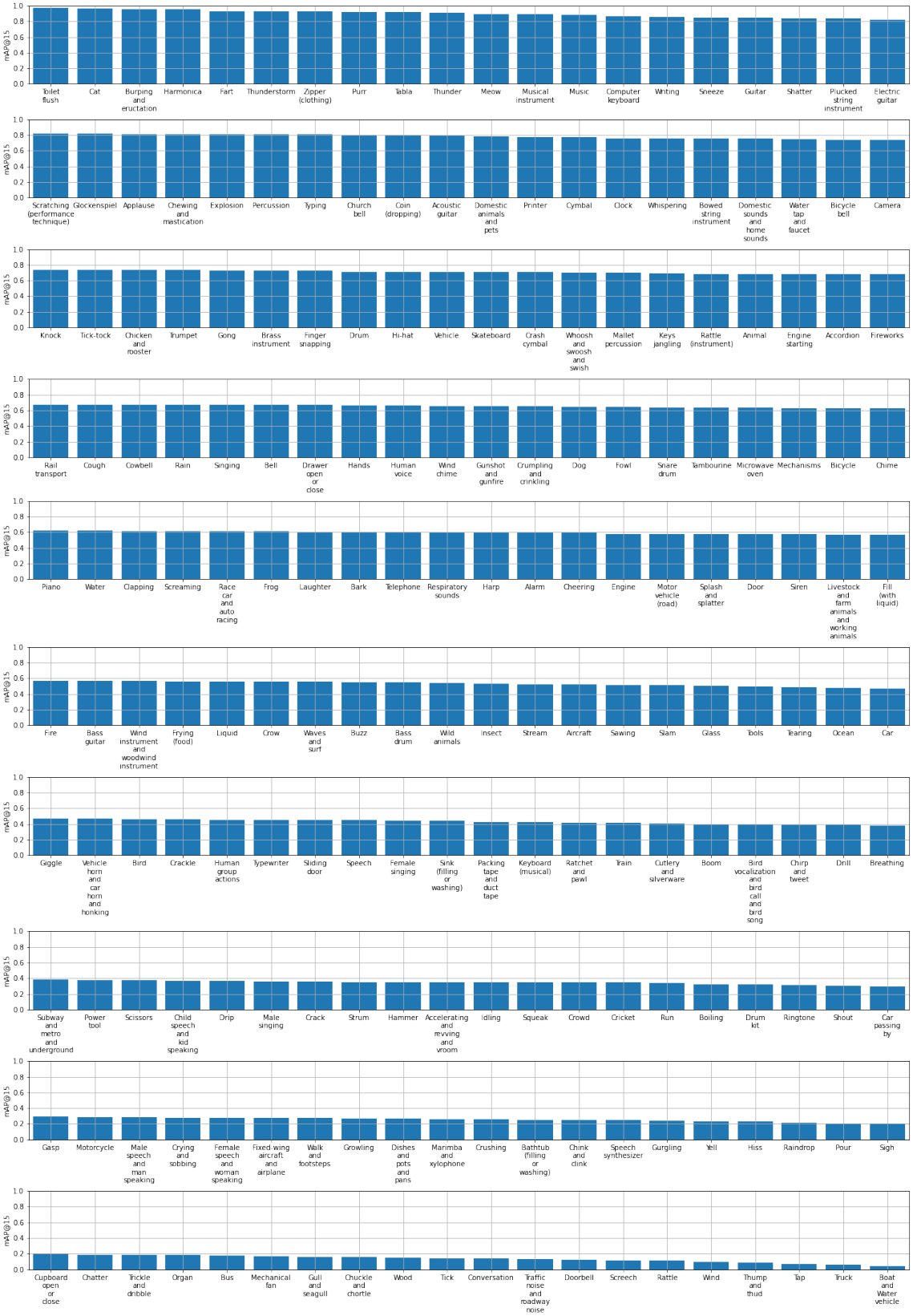

Figure 14: Semantic sound similarity performance of CLAP on sound classes by mAP@15.

4.3 Feature engineering vs deep embeddings

Results of Section 4.2 suggest that CLAP, FSD-SINet, and ImageBind models can be used to create audio representations that capture semantic sound properties. Therefore we compare the performance of Freesound with these three models.

Table 3 combines the key findings of Table 1 and Figure 11. The difference in scores with respect to each metric is clear. CLAP significantly outperforms Freesound in terms of all objective metrics and FSD-SINet and ImageBind both perform better than Freesound.

Model	mAP@15$_{total}$ (↑)	mAP@150$_{total}$ (↑)	mR1$_{total}$ (↓)
Freesound	0.09	0.03	43.2
FSD-SINet	0.33	0.18	17.7
ImageBind	0.29	0.16	21.5
CLAP	0.53	0.37	8.4

Table 3: Freesound against the top 3 deep learning based QbE systems by simple-composite performance metrics.

So far our evaluations have been based only on objective metrics, a method with intrinsic issues. As highlighted in Section 3.1.1, objective metrics are susceptible to labeling errors of the dataset. Such errors could potentially skew the metric-based assessments of model performance, hence clouding the true capabilities of the systems under evaluation.

To circumvent this issue and introduce the element of human subjectivity into our evaluation, we conducted a comprehensive subjective evaluation, using the interface described in Section 3.1.2. We performed a four-way comparison between Freesound, CLAP, FSD-SINet, and ImageBind side-by-side. The goal was to provide a deeper understanding of each system's capabilities, supplementing our previous, purely metric-based evaluations. Through the subjective evaluation, we aimed to align objective metrics with human understanding to offer a more realistic model performance.

We selected various sound classes for our experiments, including some of the best and worst-performing sound classes from Figure 14, as well as other classes chosen at random. We observed several trends in Freesound's sound recommendations:

Semantically and acoustically dissimilar. When using clip 21711, which features Chicken and rooster sounds, none of the recommended results bore any resemblance to the original query in terms of either semantic content or acoustic properties.

Acoustic Similarity Without Semantic Relevance: When using clip 71587, a Toilet flush sound, Freesound primarily recommended other sounds that are related to liquids. However, none of these recommendations actually represented the sound of a toilet flushing.

Sample Pack Bias: When querying Music instrument sounds, Freesound tended to recommend other sounds from the same sample pack.

Limited Semantic Understanding: In response to clip 112064, which features a child speaking, Freesound exclusively recommended adult voice clips.

To the same audio clips as above, the three chosen deep learning models recommended a good amount of semantically similar sounds. We believe that a real Freesound user would appreciate the recommendations. Compared to the other models, CLAP usually recommended more sounds that were distributed higher in the rankings, i.e. higher AP@15. We observed that for certain sounds (e.g. clips 58900 and 112064) ImageBind or FSD-SINet provided more accurate recommendations.

We took a closer look at sound classes where CLAP had the lowest mAP@15 scores, selecting three classes from the bottom as shown in Figure 14. We made multiple queries to CLAP for each of these sound classes. Upon listening to multiple sounds from each class, we made the following observations:

- Boat and Water vehicle: Most of the recordings contain dominant sea waves in the foreground, where the boat was hardly audible through the *hum* of its

engine. CLAP's recommended sounds for this class were all related to water or ocean sounds.

- Truck: Most of the recordings were mixtures of multiple sources, including horn sounds that dominate the clip. CLAP was able to recommend various horn sounds.

- Wind: Audioset authors note that it's a difficult sound source to record with microphones [38]. Our observations confirm this. Moreover, most of the recordings were dominated by Thunderstorm sounds. CLAP again recommended thunderstorms.

Chapter 5

Discussion

This chapter is dedicated to discussing the results of our experiments, which were reported in Chapter 4. The discussion is built around our research question which is based on the three hypotheses that were stated in Chapter 3, each section of this chapter restates a hypothesis and discusses its relevant results.

5.1 Experiments On Freesound's sound similarity

Recall $\mathcal{H}_1$:

> Freesound's sound similarity function does not capture semantic properties of sound sufficiently.

In Section 4.1, we reported the results of the conducted experiments to evaluate the sound similarity function of Freesound in terms of sound semantics. Now we discuss these results in terms of $\mathcal{H}_1$.

Figure 7 demonstrates the performance of the similar sound recommendations of Freesound for each sound class using the mAP@15 metric. Intuitively, AP is a measure of accessibility, since it combines the ratio of relevant results to the returned results with the rankings of the relevant results. With this perspective, we see that when the majority of the 200 sound classes are used as queries, the sounds in the

collection that belong to the query classes are not accessible by users. Implying the poor semantic qualities of the representations. Since it can be hard to evaluate this plot, we provide two different summaries: by using the sound class hierarchies, in Figure 8; and by simple averaging, in Table 1.

In Figure 8, we observe that the sound similarity function scores poorly for all sound families. This implies that the audio representations are not suitable for any of the considered sound families. Moreover, the fact that it performs higher for Music sounds implies that the sound representation is capturing the musical structure of sounds more than non-musical structure. Therefore, a representation that is not limited to the musical structure is needed.

The two left columns of Table 1 imply that the accessibility to similar sounds on the first page decreases towards zero on the tenth page. This implies that the probability of discovering all the relevant results is approaching zero after a certain page. Moreover, the column on the right implies that, on average, the first relevant result to a query is placed at the bottom of the 3rd page, a place where most users will not explore.

The outlined implications so far regarding Figure 7, Figure 8, and Table 1 confirm $\mathcal{H}_1$. Additional validation for this hypothesis will be supplied through a subjective evaluation in Section 5.3. A potential reason for the poor semantic sound similarity performance of Freesound can be attributed to its input representation. As discussed in Section 3.2.1, Freesound uses handcrafted features for its audio representation. Previous research indicates that accurately modeling a data distribution with such features is challenging.

5.2 Sound similarity with deep embeddings

Recall $\mathcal{H}_2$:

> General purpose deep embeddings can be used to create audio representations that capture semantic sound properties.

In Section 4.2 we reported the results of the experiments to evaluate the appropriateness of deep embeddings for semantic sound similarity. Figure 11 and Figure 12 imply together that there is a variety of deep learning models that can be used for sound similarity systems that capture broad semantic sound properties, i.e. sound classes. We are inclined to take this implication as the proof of $\mathcal{H}_2$, but we wait until discussing the results of the subjective evaluation, which will occur shortly in Section 5.3. Here we continue with discussing each model's performance in terms of semantic similarity.

As demonstrated in Figure 11 and Figure 12, OpenL3, AudioCLIP, and Wav2CLIP models show bad performance in terms of each metric. This can be attributed to their respective training objectives and the datasets on which they were trained. Specifically, OpenL3 is trained with the audio-visual correspondence task, a training objective that does not demand high-level semantic understanding of audio. Furthermore, OpenL3 versions are trained on either YouTube clips featuring musical performances or environmental sounds, both of which lack semantic understanding.

Wav2CLIP, meanwhile, derives its audio capabilities from the CLIP model, which itself lacks an understanding of audio semantics. AudioCLIP, despite its intricate multi-stage training, shows low classification accuracy on AudioSet, as detailed in Section 2.3.4, which makes us question its semantic capabilities. Consequently, we argue that the CLIP model is not suited for semantic understanding of audio.

The performance difference between YAMNet and VGGish is attributed mainly to their respective training. VGGish was trained to predict the video titles using the audio clips, while YAMNet was trained to predict the labels of the AudioSet ontology. Given that video titles are not always about the sound content, it is clear that learning audio semantics is a more appropriate goal for YAMNet.

Even with the architectural improvements of FSD-SINet, compared to YAMNet, the performance difference can be attributed to the fact that FSD-SINet was trained on the FSD50K development set. Although the development and evaluation sets were split carefully to eliminate data leakage, there are similarities between the

development and evaluation set [16]. Therefore, a separate evaluation set is needed to explain this performance difference.

Since ImageBind is an unsupervised model, it might come surprising at first that its embeddings performed worse than the embeddings of FSD-SINet, a supervised model. We attribute this difference to the training objective of ImageBind. Aligning sound-image pairs and text-image pairs while not aligning sound-image pairs results in less semantically meaningful audio embeddings. Also, the use of depth and motion modalities may have reduced the semantic properties of the embeddings. Lastly, while FSD-SINet utilized the FSD50K dataset for training, ImageBind was trained on AudioSet, contributing to their performance difference.

Finally, CLAP outperforms every other model in terms of all metrics. We attribute this advantage to its training objective, which utilizes natural language supervision for sounds. The model is trained on a dataset that includes examples from the FSD50K development set. Importantly, these Freesound clips include textual descriptors detailing the recording conditions, which are preserved in CLAP's training objective. Contrary to our belief, these textual descriptions do not appear to have compromised the model's capacity for semantic understanding of audio. The additional training data may have compensated for this possibly negative effect.

5.3 Feature engineering vs deep embeddings

Recall $\mathcal{H}_3$:

> For semantic sound similarity, audio representations obtained from deep embeddings can work better than Freesound's audio representation.

In the subjective evaluation test detailed in Section 4.3, we found that CLAP consistently yielded more pertinent information that was ranked higher in the sequence compared to FSD-SINet, which itself ranked higher than ImageBind, and in turn, higher than Freesound. Notably, this sequence aligns perfectly with the objective performance metrics laid out in Table 3. This alignment between our subjective

and objective evaluations in Table 3 has two implications: First, it substantiates that the Average Precision (AP) and Rank-1 (R1) metrics we employed are strongly correlated with human judgment. Second, it reinforces the conclusion that among the models we tested, CLAP stands out as the most proficient in capturing semantic similarity.

Now that we validated the effectiveness of the objective evaluations, we can conclude that $\mathcal{H}_2$ is validated. Moreover, together with the discussion relating the Table 3, we consider $\mathcal{H}_3$ validated as well.

We close this chapter by stating that our research question of whether Freesound's sound similarity function can be improved in terms of semantics is answered by proving the three hypotheses that were discussed in this chapter.

Chapter 6

Conclusion

In this work, we explored various techniques for modeling the semantic properties of sound with the purpose of being integrated into a sound similarity function of an audio information retrieval system, Freesound. Our work yielded several findings that contribute to the field of sound similarity.

We established that hand-crafted acoustic features are insufficient for capturing the semantic contents of audio, especially when lacking specialized processing. On the contrary, we demonstrated that general-purpose deep learning models can be easily adapted to create audio representations that are rich in semantic content. Through comparisons and analyses, we explored the impact of various design choices regarding the processing of deep embeddings for audio representation and identified optimal algorithms for similarity search.

Our investigation included both supervised and unsupervised learning methods, leading us to identify a number of effective methods for creating audio representations rich in semantic content. In a noteworthy finding, unsupervised models that use only natural language and audio as modalities were shown to outperform both supervised methods and unsupervised methods that use non-language modalities.

We also examined models designed for specific tasks, such as audio-visual correspondence, and models that attempt to leverage the CLIP model for audio representation.

Our analyses showed that these models are suboptimal for capturing the semantic properties of audio. Specifically, we observed the shortcomings of such models in their ability to understand audio semantics.

Our work is the validation of the CLAP model, which consistently outperformed all other models across various evaluation metrics. Through objective and subjective evaluations, we established that the CLAP model provides superior semantic representations, and also that the metrics used in our evaluations correlate well with human judgments. This dual validation highlights the robustness and reliability of our findings.

In summary, this study shows the limitations of the traditional, feature selection-based sound similarity approaches, the availability of the general purpose audio representations for semantic sound similarity systems, and specifically the usefulness of the representations obtained from natural language supervision. We hope that our findings will increase Freesound's value for the community.

By combining deep learning techniques, information retrieval methods, and a diverse set of evaluation methods, we have contributed to a more comprehensive understanding of audio semantics for sound similarity.

Bibliography

[1] Font, F. *Design and evaluation of a visualization interface for querying large unstructured sound databases*. PhD Thesis, Universitat Pompeu Fabra (2010). URL https://doi.org/10.5281/zenodo.1173914.

[2] Favory, X. *Improving Sound Retrieval in Large Collaborative Collections*. PhD Thesis, Universitat Pompeu Fabra (2021). URL http://hdl.handle.net/10803/671207.

[3] Chechik, G., Ie, E., Rehn, M., Bengio, S. & Lyon, D. Large-scale content-based audio retrieval from text queries. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 105–112 (ACM, Vancouver British Columbia Canada, 2008). URL https://dl.acm.org/doi/10.1145/1460096.1460115.

[4] Barrington, L., Chan, A., Turnbull, D. & Lanckriet, G. Audio Information Retrieval using Semantic Similarity. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, II–725–II–728 (IEEE, Honolulu, HI, 2007). URL https://ieeexplore.ieee.org/document/4217511/.

[5] Mechtley, B., Wichern, G., Thornburg, H. & Spanias, A. Combining semantic, social, and acoustic similarity for retrieval of environmental sounds. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2402–2405 (IEEE, Dallas, TX, USA, 2010). URL http://ieeexplore.ieee.org/document/5496225/.

[6] Wu, Q., Zhang, X., Lv, P. & Wu, J. Perceptual similarity between audio clips and feature selection for its measurement. In *2012 8th International Symposium on Chinese Spoken Language Processing*, 387–391 (IEEE, Kowloon Tong, China, 2012). URL http://ieeexplore.ieee.org/document/6423476/.

[7] Yu, X., Zhang, J., Liu, J., Wan, W. & Yang, W. An audio retrieval method based on chromagram and distance metrics. In *2010 International Conference on Audio, Language and Image Processing*, 425–428 (IEEE, Shanghai, China, 2010). URL http://ieeexplore.ieee.org/document/5684543/.

[8] Manocha, P. *et al.* Content-Based Representations of Audio Using Siamese Neural Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3136–3140 (IEEE, Calgary, AB, 2018). URL https://ieeexplore.ieee.org/document/8461524/.

[9] Sert, M. & Basbug, A. M. Combining Acoustic and Semantic Similarity for Acoustic Scene Retrieval. In *2019 IEEE International Symposium on Multimedia (ISM)*, 156–1563 (IEEE, San Diego, CA, USA, 2019). URL https://ieeexplore.ieee.org/document/8959049/.

[10] Fan, J. *et al.* Multi-Label Sound Event Retrieval Using A Deep Learning-Based Siamese Structure With A Pairwise Presence Matrix. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3482–3486 (IEEE, Barcelona, Spain, 2020). URL https://ieeexplore.ieee.org/document/9053972/.

[11] Hershey, S. *et al.* CNN Architectures for Large-Scale Audio Classification (2017). URL http://arxiv.org/abs/1609.09430. ArXiv:1609.09430 [cs, stat].

[12] Gong, Y., Chung, Y.-A. & Glass, J. AST: Audio Spectrogram Transformer (2021). URL http://arxiv.org/abs/2104.01778. ArXiv:2104.01778 [cs].

[13] Fonseca, E., Ferraro, A. & Serra, X. Improving Sound Event Classification by Increasing Shift Invariance in Convolutional Neural Networks (2021). URL http://arxiv.org/abs/2107.00623. ArXiv:2107.00623 [cs, eess].

[14] Wu, H.-H., Nieto, O., Bello, J. P. & Salamon, J. Audio-Text Models Do Not Yet Leverage Natural Language (2023). _eprint: 2303.10667.

[15] Keivani, O., Sinha, K. & Ram, P. Improved maximum inner product search with better theoretical guarantee using randomized partition trees. *Machine Learning* **107**, 1069–1094 (2018). URL http://link.springer.com/10.1007/s10994-018-5711-7.

[16] Fonseca, E., Favory, X., Pons, J., Font, F. & Serra, X. FSD50K: An Open Dataset of Human-Labeled Sound Events (2022). URL http://arxiv.org/abs/2010.00475. ArXiv:2010.00475 [cs, eess, stat].

[17] Gemmeke, J. F. *et al.* Audio Set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 776–780 (IEEE, New Orleans, LA, 2017). URL http://ieeexplore.ieee.org/document/7952261/.

[18] Dupret, G. Discounted Cumulative Gain and User Decision Models. In Grossi, R., Sebastiani, F. & Silvestri, F. (eds.) *String Processing and Information Retrieval*, vol. 7024, 2–13 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2011). URL http://link.springer.com/10.1007/978-3-642-24583-1_2. Series Title: Lecture Notes in Computer Science.

[19] Zhang, E. & Zhang, Y. Average Precision. In LIU, L. & ÖZSU, M. T. (eds.) *Encyclopedia of Database Systems*, 192–193 (Springer US, Boston, MA, 2009). URL https://doi.org/10.1007/978-0-387-39940-9_482.

[20] Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* (MIT Press, 2016).

[21] Jansen, A. *et al.* Unsupervised Learning of Semantic Audio Representations (2017). URL http://arxiv.org/abs/1711.02209. ArXiv:1711.02209 [cs, eess, stat].

[22] Sound classification with YAMNet | TensorFlow Hub. URL https://www.tensorflow.org/hub/tutorials/yamnet.

[23] Kong, Q. *et al.* PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition (2020). URL http://arxiv.org/abs/1912.10211. ArXiv:1912.10211 [cs, eess].

[24] Gong, Y., Chung, Y.-A. & Glass, J. PSLA: Improving Audio Tagging with Pretraining, Sampling, Labeling, and Aggregation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **29**, 3292–3306 (2021). URL http://arxiv.org/abs/2102.01243. ArXiv:2102.01243 [cs, eess].

[25] Verma, P. & Berger, J. Audio Transformers:Transformer Architectures For Large Scale Audio Understanding. Adieu Convolutions (2021). URL http://arxiv.org/abs/2105.00335. ArXiv:2105.00335 [cs, eess].

[26] Elizalde, B., Deshmukh, S., Ismail, M. A. & Wang, H. CLAP: Learning Audio Concepts From Natural Language Supervision (2022). URL http://arxiv.org/abs/2206.04769. ArXiv:2206.04769 [cs, eess].

[27] Saeed, A., Grangier, D. & Zeghidour, N. Contrastive Learning of General-Purpose Audio Representations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3875–3879 (IEEE, Toronto, ON, Canada, 2021). URL https://ieeexplore.ieee.org/document/9413528/.

[28] Arandjelović, R. & Zisserman, A. Look, Listen and Learn (2017). URL http://arxiv.org/abs/1705.08168. ArXiv:1705.08168 [cs].

[29] Cramer, A. L., Wu, H.-H., Salamon, J. & Bello, J. P. Look, Listen, and Learn More: Design Choices for Deep Audio Embeddings. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3852–3856 (IEEE, Brighton, UK, 2019). URL https://ieeexplore.ieee.org/document/8682475/.

[30] Favory, X., Drossos, K., Virtanen, T. & Serra, X. COALA: Co-Aligned Autoencoders for Learning Semantically Enriched Audio Representations (2020). URL http://arxiv.org/abs/2006.08386. ArXiv:2006.08386 [cs, eess, stat].

[31] Wu, H.-H., Seetharaman, P., Kumar, K. & Bello, J. P. Wav2CLIP: Learning Robust Audio Representations From CLIP (2022). URL http://arxiv.org/abs/2110.11499. ArXiv:2110.11499 [cs, eess].

[32] Guzhov, A., Raue, F., Hees, J. & Dengel, A. AudioCLIP: Extending CLIP to Image, Text and Audio (2021). URL http://arxiv.org/abs/2106.13043. ArXiv:2106.13043 [cs, eess].

[33] Girdhar, R. *et al.* ImageBind: One Embedding Space To Bind Them All (2023). URL http://arxiv.org/abs/2305.05665. ArXiv:2305.05665 [cs].

[34] Bogdanov, D. *et al.* ESSENTIA: an Audio Analysis Library for Music Information Retrieval. In *Proceedings - 14th International Society for Music Information Retrieval Conference* (2013).

[35] Analysis Descriptor Documentation — Freesound API documentation. URL https://freesound.org/docs/api/analysis_docs.html.

[36] Apache Solr Reference Guide :: Apache Solr Reference Guide. URL https://solr.apache.org/guide/solr/latest/index.html.

[37] models/research/audioset/yamnet at master · tensorflow/models. URL https://github.com/tensorflow/models.

[38] AudioSet. URL http://research.google.com/audioset/.