JOHANNES KEPLER
UNIVERSITY LINZ

Author
DI **Nikita Tymofieiev**, BSc
12133178

Submission
Institute of
Computational Perception

Thesis Supervisor
Assistant Prof. **Paul Primus**

October 2025

Cross-Modal Text-Audio Retrieval: Experiments

Bachelor's Thesis

to confer the academic degree of

Bachelor of Science

in the Bachelor's Program

Artificial Intelligence

JOHANNES KEPLER
UNIVERSITY LINZ
Altenberger Straße 69
4040 Linz, Austria
jku.at

Abstract

This work studies text-to-audio retrieval, which plays an important role in multimedia research. The goal of this task is to learn audio and text representations such that matching items from different modalities lie close in the embedding space and have high similarity. On the user side, audio samples can be retrieved using a text query and vice versa. The Sigmoid Loss designed for image-language pretraining is compared to NT-Xent. Another series of experiments uses a transformer-based ATST audio encoder and shows a 2.41 pp. mAP@10 gain after reducing its learning rate. Furthermore, the Text-Aware Attention module, which is essentially a scaled dot-product attention mechanism that allows text to attend to its most semantically similar frames, leads to a moderate improvement—on average yielding an additional 0.65 pp. mAP@10. Finally, the analysis of lexical richness indicates that human captions exhibit greater diversity than generated ones.

Contents

Chapter 1

Introduction

Text to Audio Retrieval (TAR) refers to the task of retrieving audio files that are relevant to a given natural language query. Upon a user's request, a system will rank audio according to a similarity metric and retrieve matching results from a corpus of audios in order of relevance to a given query. This corpus may include, but is not limited to, a podcast, a sound library, or an audio of a video. This may serve both educational and practical purposes. Its application use cases may include Multimedia Search Engines, where a user could provide a textual description of a sound that needs to be found via an intuitive interface. Text-to-audio retrieval is difficult because it requires aligning two very different modalities—language and sound—into a shared semantic space despite their vastly different structures and ambiguities.

The early work focused on retrieving audio based on its metadata which was not exactly audio-content based [2]. Cross-modal tasks (e.g. image-text retrieval) have received enormous attention [33]. However, not as much attention was paid to this task in the saudio domain. With recent advancements and development of audio-captioning datasets [7, 16, 25] and success of cross-modal tasks in vision domain, text-audio retrieval has been gaining traction.

Current state-of-the-art models use dual-encoder system, where audio and text are encoded in a shared embedding space. As an audio encoder, CNN-based [18] and Transformer-based [19, 21] architectures are commonly employed, while BERT [5] and RoBERTa [22] are regularly used as text encoders. In this work, audio spectrogram transformer PaSST [19] and ATST [21] along with RoBERTa are used.

Experiment section explores benefits of different learning rates of encoders of different modalities. It is shown that the Text Aware Pooling module, an aggregation technique presented in [43] that attends over most semantically audio frames to a given text, leads to consistent but more modest improvement.

Another attempt that has been made to improve performance is data augmentation by Large Language Model generated captions. In [29] it was shown, that it results in quite incremental gains. A deeper insight is found by comparing lexical richness applied to both original and generated captions.

Implementation of experiments as well as generated captions are publicly available in a GitHub repo.[1].

[1] https://github.com/timoniko/PR-Text-To-Audio-Retrieval

Chapter 2

Background and Related Work

2.1 First Stages

The problem of large-scale cross-modal retrieval dates back to [2]. This work was the first to enable retrieval based solely on acoustic content without relying on additional textual metadata. It operates using traditional machine learning techniques such as Support Vector Machines and Gaussian Mixture Models by learning to match sounds to text tags using Mel-Frequency Cepstral Coefficients as audio high-level features. In this paper, a scoring function is learned to rank audio samples according to its associated textual tag. A constraint of this study is that the labels utilized in the query must precisely correspond to those in the index. Such a system cannot generalize to unseen words or slightly different phrasing. Another early work [37] proposed a more flexible approach where new sounds could be associated with existing labels. The authors used a hierarchical language model which limits the scalability. Neither of these approaches enables a direct comparison between text-audio and audio-text pairs nor do they learn to map similarities between textual semantics and acoustics. One of the more recent [9] works addresses this issue by proposing a framework that learns joint embeddings from a shared lexico-acoustic space, such that vectors of either modality can be compared directly. The main limitation of these methods is that they are designed to handle single-word queries rather than free-form natural language.

Lately, the very first benchmark was established in [17]. In this paper, pretrained models and common ideas from video retrieval have been adopted to address the scarcity of audio-text data. The turning point was the result from [24], which presented a full free-form language-based audio-text retrieval model and studied various learning metrics for model training. Modern cross-modal retrieval tasks are typically framed as contrastive learning problems.

2.2 Contrastive Learning

Contrastive learning is a training approach in which a model learns to differentiate between similar and dissimilar data points. It aims to create representation of data where similar instances are close together in the embedding space, while dissimilar instances are far apart. This is done via minimizing the distance between similar instances (*positive pairs*) and maximizing the distance between dissimilar pairs (*negative pairs*). An *anchor* is a reference sample a model learns from used to created positive and negative pairs for training. This learning approach has gained a massive importance in different fields of Deep Learning. Although research in this area primarily focused on Computer Vision, this learning paradigm is as well applicable to tasks in Audio and Natural Language Processing.

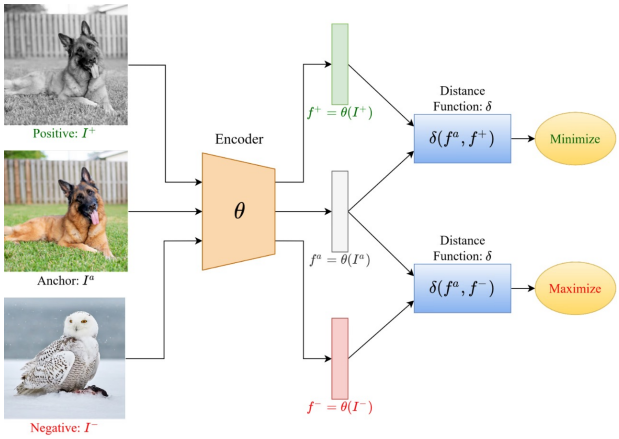


Figure 2.1: Self-supervised contrastive learning pipeline. A model learns an embedding function to maximize similarity between an anchor(middle left image) and its augmented version - a positive pair(top left image), while minimizing its distance with a negative pair(bottom left image). (from [20])

Supervised Contrastive Learning (SCL) [15] operates on manually annotated data. The disadvantage of this procedure lies within its need for carefully constructed labels.

Self-Supervised Contrastive Learning (SSCL) [3] is of main interest in context of text to audio retrieval. The general distinction from SSL is that it makes use of unlabeled data, i.e. there are no predefined classes. A common technique in Computer Vision is creating positive instances for an anchor sample by means of augmentation techniques such as cropping, color transformation, flipping and rotation. Negatives consist then of the entire remainder of a batch. Figure 2.1 provides an illustration.

2.3 CLIP: Contrastive Language-Image Pretraining

In computer vision, state-of-the art classification systems are trained on a predefined set of categories. This poses challenges, since dataset annotation is costly and labor intensive. As an example, ImageNet [4] dataset required approximately 25,000 workers to annotate more than 14 million images across about 22,000 categories during its initial large-scale annotation phase. The generalization to unseen classes is often achieved via *fine-tuning*, by unfreezing an output head of a pretrained model. Yet, standard vision approaches struggle to classify beyond provided labels, significantly narrowing down the number of visual concepts that can be learned.

One of the more prominent frameworks aimed at mitigating this issue is CLIP (Contrastive Language-Image Pretraining) [33] which was introduced by OpenAI in 2021. CLIP is a multimodal neural network that efficiently learns visual concepts from natural language supervision, enabling *zero-shot classification*, which can be considered as instance of *transfer learning*. It builds on a large corpus of work related to multimodal learning, zero-shot transfer, and natural language processing.

To make manual labeling independence feasible, 400 million image-text pairs were scraped from various internet sources to form a WebImageText (WIT) dataset (not publicly available). The dataset is *weakly-labeled*, as the pairs are not manually curated and labels may vary in length, quality and relevance.

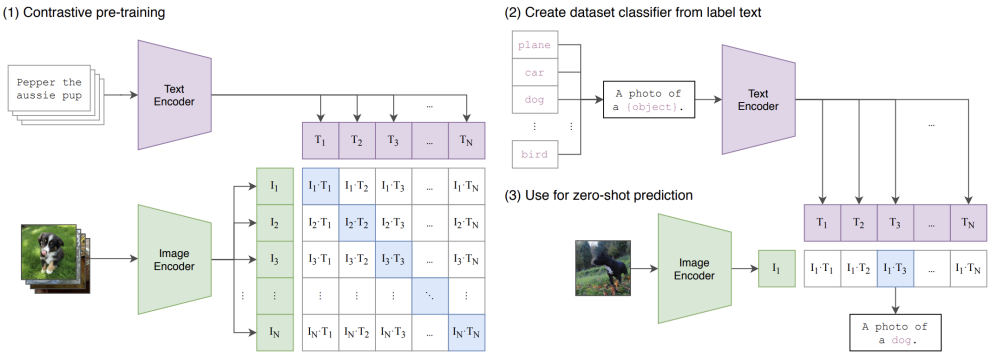


Figure 2.2: The dual encoder system is used to project text and audio pairs in multi-modal embedding space. Then, a mini-batch is formed by computing cosine similarities between all possible pairs. (from [33])

At training and prediction, the output of audio and text encoders are vectors of length 1024. The main idea is computing similarities not just between corresponding images and captions, but between all image-captions in the batch when training a model. Aligning these pairs is a multimodal contrastive task. Given a batch of N (image, text) pairs, CLIP is trained to predict which of the $N \times N$ possible (image, text) pairings across the batch actually occurred. To do this, a multimodal embedding space is learned by jointly training an image and text encoders to maximize cosine similarity between matching images and texts.

At inference time, zero-shot classification is enabled, allowing to transfer knowledge from the pretrained model to an unseen task. This way, similar accuracy to a ResNet-50 trained on ImageNet was achieved — but without training on a single ImageNet image.

CLIP laid a significant conceptual groundwork for cross-modal learning and contrastive pretraining, influencing development of models adopting these principles to an audio domain.

2.4 Learning audio concepts from natural language

In recent years, a significant progress been made in audio domain. As in computer vision, mainstream audio models are trained under a supervised learning paradigm, where a predefined label is assigned to each recording, which can be a sound event ot acoustic scenes. This approach limits generalizing to unseen classes. Computer Vision has successfully adopted learning visual representations with natural language supervision. One of more influential examples is early mentioned CLIP model.

2.4.1 AudioCLIP

AudioCLIP [14] is extension of CLIP that incorporates an audio encoder into the CLIP model, thus obtaining a *tri-modal* architecture. The model employs contrastive learning to perform training on textual, audio and image data. This approach allows classification and cross-modal querying using text, image and audio in any combination. AudioCLIP model incorporates three subnetworks: text-, image- and audio-heads. Alongside the current text-to-image similarity loss term, two additional terms have been introduced: text-to-audio and image-to-audio.

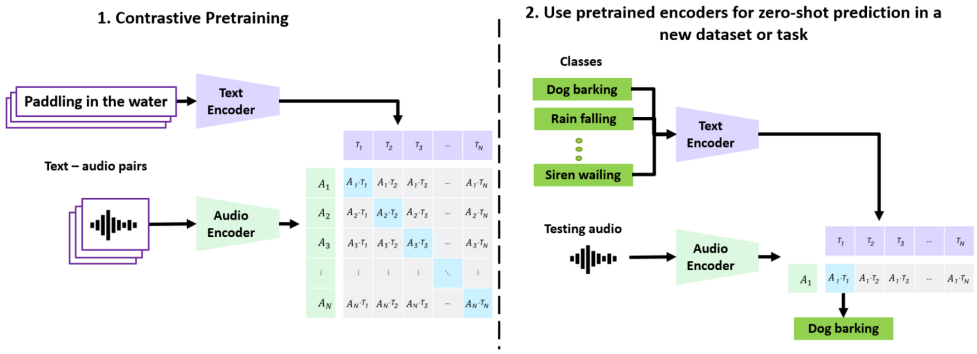


Figure 2.3: CLAP training workflow and zero-shot inference. (from[8])

This work uses five image, audio and mixed datasets which were used directly and indirectly. The audio encoder is initialized with ImageNet [4] weights. This includes the composite CLIP dataset, which was used indirectly as weights initializer. To enable working with three modalities, the main dataset for training was AudioSet [10]. Each sample is a snippet up to 10 seconds long from a YouTube-video, defined by the corresponding ID and timings. During training step, image is chosen randomly from ten equally distributed frames. The model is capable of doing both classification and querying(retrieval) in any direction.

AudioCLIP is an important precursor to paired audio–text contrastive models like CLAP [8], because it demonstrates that large-scale image–text models can bootstrap an audio–text space. The main limitation is that it relies on label-based supervision rather than natural-language captions, which limits the expressiveness and flexibility of retrieval. Additionally, it uses the frozen text encoder from CLIP, which may not semantically align with audios.

2.4.2 CLAP: Contrastive Language-Audio Pretraining

Contrastive Language-Audio Pretraining (CLAP) [8] appears to be the first model to fully bring natural language supervision for learning audio concepts. CLAP is a dual-encoder neural network trained with 128000 audio and text pairs from four different datasets, including Clotho [7]. 16 datasets from 8 domains were used as downstream tasks for classification, outperforming supervised setups at the moment of publication. The workflow is shown on Figure 2.3.

Analogous to CLIP, the training data of CLIP consists of description containing one or more sentences. In classification, however, labels are usually denoted with a single word or a few words such as "dog barking" and "sneezing". A method to address this distribution discrepancy is using a prompt template. Examples of such prompt templates are `{this is an audio of {class label}}` or `{this is a sound of {class label}}`. Choosing a proper template leads to an improved performance.

CLAP was not the first attempt of learning audio by means of natural language, but it was the first major large-scale, end-to-end pretrained model trained with a contrastive loss between audio and text.

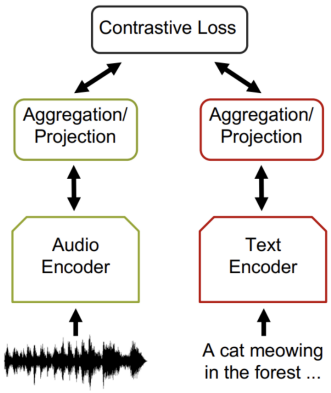


Figure 2.4: Training Overview for a Dual-Encoder Text–Audio Retrieval System (from [41])

2.5 Modern Text-to-Audio Retrieval

Previously described models were innovative in enabling large-scale cross-modal learning between image–text and audio–text modalities. However, their primary focus was not on retrieval as a task in itself, but rather on zero-shot classification, with evaluations based on classification-oriented metrics.

Text-audio retrieval task is different from classification. In classification, only the label with a highest similarity score is predicted. In TAR, a performance is measured by Recall (*R@10, R@5, R@1)* and Mean Average Precision (*mAP@10*). Similarly to CLAP, TAR systems follow a general pipeline with bi-encoder architecture trained with contrastive loss between audio and texts as shown in Figure 2.4. It is one of the tasks featured in the Detection and Classification of Acoustic Scenes and Events (DCASE) competition that has been featured since 2022 [38] and has made a rapid progress since.

2.5.1 Datasets

The three widely adopted datasets are AudioCaps [16], WavCaps [25], and Clotho [7].

Clotho V2 is a publicly available audio captioning dataset built with focus on audio quality and caption diversity. It contains 15-30 second recordings and captions that are between 8 and 20 words long. The provided training, validation, and test split contain 3840, 1045, and 1045 recordings, respectively; each recording is associated with five human-generated captions. The dataset has served as the benchmark of retrieval models in task 6B of DCASE challenge.

AudioCaps consists of 51308 audio recordings taken from AudioSet [10]. Each training and validation recording is associated with one and five human-written captions, respectively. The audio recordings' length is roughly 10 seconds, and the captions are 9.8 words long on average. AudioCaps has been actively used to improve performance on Clotho.

WavCaps is a large weakly-labeled audio captioning dataset encompassing approximately 400k audio clips with paired captions. The data is collected from web platforms-Freesound, BBC Sound Effects, and SoundBible, as well as a subset of audio tagging dataset, AudioSet. Some of these textual annotations are noisy or do not describe a sound at all, capturing unrelated information, such as names of recording devices, time, or locations. To alleviate the problem of low-quality captions, it underwent three-stage preprocessing pipeline:

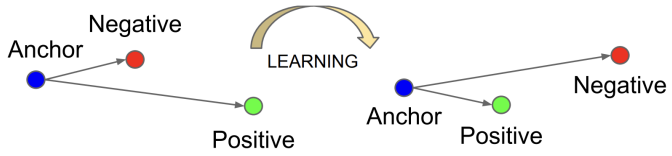


Figure 2.5: The Triplet Loss encourages the model to minimize the distance between an anchor and a positive sample sharing, while maximizing the distance between the anchor and a negative sample. (from[35])

1. **Pre-Filtering** consists of audio-duration filtering and high-frequency text filtering. For audio-duration filtering, audio clips shorter than one second are removed. High-frequency text filtering is applied to exclude descriptions that are shared by more than 5 audio clips.
2. **ChatGPT-based Transformation** is used to transform audio tags into caption-resembling sentences and to remove redundant information. Despite its efficiency, it may still fail to follow prompts in a small percentage of tasks requested including geographic locations, names and numbers.
3. **Post-Processing** addresses aforementioned cases of incorrect processing by utilizing an entity recognition library. Any caption with a named entity is then again transformed with ChatGPT until all captions follow an instruction.

WavCaps is commonly paired with AudioCaps because of complementary strength of each dataset, with WavCaps significantly improving performance before fine-tuning on Clotho, as was shown in [29], which investigated benefits of large audio-captioning datasets.

2.5.2 Metric learning objectives

Triplet Loss

The Triplet Loss emerges from the field of Face Recognition [35]. Triplet-based loss functions rely on the concept of a triplet, which consists of an anchor, a positive (paired candidate in another modality) and a negative (unpaired candidate in another modality) as visualized on Figure 2.5

Triplet-sum loss aims to maximize the similarity score of its positive pair while minimizing the similarity scores to all other negatives within a mini-batch. This however can lead to easy negatives dominating the loss resulting in loss getting stuck at a global minimum. **Triplet-max** loss focuses on the hardest negatives during training by maximizing the similarity score of its positive pair while just minimizing the similarity score to its hardest negative within a mini-batch. **Triplet-weighted** loss also focuses on hard negatives, but does not weight positives and negatives equally. It was shown in [24] that hard negative mining variants perform better, although they need more careful tuning.

InfoNCE

The InfoNCE Loss, where "NCE" stands for *Noise-Contrastive Estimation* [13], was presented in [39] in context of learning useful representations from high-dimensional data in unsupervised learning. The purpose is to maximize the lower bound of mutual information. Since there are no "current" or "future" samples in a pure contrastive learning

setting, but rather positive and negative pairs, multimodal learning literature adheres to notation with batch-wise CL and matrix style notation (eq.3.1). InfoNCE loss is often called *cross-entropy loss* in context of CL.

NT-Xent Loss

The NT-Xent loss (normalized temperature-scaled cross-entropy loss) (eq. 3.2) comes from SimCLR [3] (*Contrastive Learning of Visual Representations*) paper. SimCLR learns representations by maximizing agreement between two differently augmented views of the same data example via a contrastive loss. In the paper, one view serves as target for another view of the same image, while agreements to any other images are minimized, producing a matrix of size $2N$.

It is the special case of InfoNCE (eq. (3.1)), as it also computes *Cross Entropy* for each row of similarity matrix, aligning the model's predicted similarity distribution over one positive against all negatives via a softmax. The main difference lies in symmetricity. That is, the final loss is computed across all positive pairs, both (i, j) and (j, i) in a mini-batch. When implementing the bidirectional InfoNCE (text $\rightarrow$ audio and audio $\rightarrow$ text) they become mathematically identical. In [24], which studied the impact of different metric learning objectives on the audio-text retrieval task, NT-Xent Loss was found to outperform triplet-based losses and has been used almost exclusively for this task.

Sigmoid Loss

Sigmoid Loss (eq. 3.3) appeared in [47] as an alternative loss for Language Image Pre-training, arguing that a widespread implementation of softmax-based contrastive loss (e.g. NT-Xent) is numerically unstable, as it requires global view of the pairwise similarities for normalization. Each text-image pair is processed independently and treated as a binary classification task. It allows for more efficient training, while also showing better performance at smaller batch sizes. To the current moment, no prior work tested this loss in text-to-audio retrieval task.

2.5.3 DCASE Challenge & Broader Research

DCASE 2022

Leading Submission

As the challenge started in 2022, text-to-audio retrieval and audio-captioning were combined in a single task. Xu et al. [45] experimented with CNN14 and Wavegram-Logmel-CNN14 in PANNs [18] and AST [11] and performed AudioCaps pretraining before fine-tuning on Clotho. PANNS (Pretrained Audio Neural Networks) is a family of deep learning models designed for general-purpose audio classification and transfer learning tasks. These models were trained on the large-scale AudioSet dataset. AST, or Audio Spectrogram Transformer, is a deep learning model that applies the Vision Transformer (ViT) [6] architecture to audio classification tasks. AST performed worse than CNN-based methods. BERT and RoBERTA were tested as text encoders. RoBERTA outperformed BERT and has been a dominant choice for learning text representation in this task. This work also demonstrated that an ensemble of models with different architectures significantly improves performance.

Some other works including the baseline leveraged word2vec [27], an older and popular technique in natural language processing used to map words to numerical vectors, which was outperformed by transformer-based architectures later on.

Contemporary & Follow-up Contributions

Many systems used SpecAugment [28]. SpecAugment is a data augmentation method for audio, especially used in speech and audio classification tasks. The processes include time warping, frequency masking, and time masking. However, not as much attention was paid to text augmentations. Arguing that publicly available audio-captioning datasets are relatively small compared to those in image domain, Primus and Widmer [32] employ various experiments with various data augmentation techniques and show that this reduces overfitting and improves retrieval performance. Along with SpecAugment, random gain augmentation and domain shift simulation techniques were enabled. Word-level modification and back-translation were employed for the text. Text augmentation had a greater effect than audio augmentation, but both types of augmentation demonstrated improved performance. Additionally it was found that pretraining on AudioCaps did not lead to very impactful improvement.

Another study by Wu et al. [41] concluded that existing models do not leverage natural language and advocate for collection of more diverse data. They conducted a series of experiments and found that

1. current models focus on nouns and verbs without utilizing an entire sentence.
2. current models cannot capture temporal and sequential events.
3. unlike previous state-of-the art systems, a proposed transformer-based architecture is capable to capture sequential events.
4. existing benchmarks are insufficient for evaluating the sequential modeling abilities of audio-text models.

A similar problem but for audio was tackled by Xin et al. [43]. The semantic information contained in the text is only similar to certain frames within the audio. It is argued that existing models aggregate the entire audio without considering the text and aggregation techniques such as mean pooling may encode misleading audio information not described in a text. Authors propose a **Text-Aware Attention Pooling** (also referred to as **TAP**), a module that is essentially a scaled dot product attention for a text to attend to its most semantically similar frames. In this work TAP proved to outperform text-agnostic pooling functions such as mean pooling.

DCASE 2023

Leading Submission

The next year, WavCaps was introduced which set a new state-of-the art on ClothoV2 dataset. Unlike most submissions last year that used CNN-based architectures, Primus et al. [29] used PaSST, a transformer-based audio encoder. PaSST employs Patchout during training, which increases training speed and memory efficiency while at the same time acting as a regularizer. Its pretrained parameters are taken from Vision Transformer which is trained on ImageNet. PaSST was subsequently fine-tuned on AudioSet for general-purpose audio tagging. Pretraining on both WavCaps and AudioCaps resulted in significantly better representation space. This work also studied the effect of augmenting existing captions with synthetic ones using a Large Language Model (ChatGPT) via

API and found that it leads to a slightly better performance. As in the previous year, an ensemble of different models resulted in significantly better retrieval.

Contemporary & Follow-up Contributions

A overlooked problem of text-based audio retrieval is a language limitation. Yan et al. [46] argue that predominant focus of text-based retrieval on English poses a big limitation on the applicability of these models. Their approach involved using a multilingual text decoder to encode the text data with language-specific information. A multilingual text translator was employed to translate the English descriptions from the training set into seven additional languages. One caption of five per audio of Clotho dataset is randomly selected into seven different languages which forms multilingual audio-text pairs using the same audio. In addition to improved proficiency of retrieval in new languages, their chosen audio and text encoders also show consistent gains in performance.

A problem of data scarcity was addressed in [44]. Unlike in image-language tasks, audio-language learning faces challenges due to limited and lower-quality data compared to image-language tasks. Authors introduce AudioSetMix, an audio-captioniong dataset generated through the application of audio transformations to clips from AudioSet. The idea of turning labels into natural language descriptions is not unheard of (e.g. WavCaps). AudioSetMix incorporates both audio and text augmentations. In their approach, audios undergo operations such as volume change, pitch and duration change with a certain probability. Moreover, they define concatenation and mix of two audio clips. Every transformation plays a role in further caption generation. More precisely, they use LLM to generate natural language description of the new audio clips based on the augmentations applied in the previous step. Newly generated captions may describe new temporal events in addition to emphasizing other changes such as volume. Inspired by [41], it was investigated whether audio-language models also fail to "understand" the modifiers. Only amarginal difference in performance was noticed if replacing modifiers by antonyms (e.g. *loud* $\rightarrow$ *quiet*) in AudioCaps and Clotho which further proves main point in [41]. However, retrieval still benefited substantially from additional 132k training audio-text pairs from AudioSetMix.

Another work by Primus and Widmer [31] explores usage of metadata such as key words which are often attached to audios and find that it improves retrieval. Using metadata to generate artificial natural language captions with an LLM is not a new concept. This however completely neglects the metadata during inference while also inflating training data. The proposed methodology involves creating audio-metadata pairs by encoding audio and its metadata in separate embedding representations and then fusing them into a single item before a text query.

DCASE 2024

Leading Submission

Primus et al. [29] argue that one of the main challenges of datasets in text-based audio retrieval is the assumption that a caption can correspond to a single audio only. In other words, "CLAP-like" systems assume strictly binary relevance between audios and captions. However it is likely that audio-caption datasets contain semantically similar captions for different audio samples. For instance, captions *"A cat is crying and a person is speaking"* and *"A man is talking and a cat crying"* are semantically similar, yet they come from two different audios (example from [42]). This will lead to false negatives if both appear in a batch.

Previous studies assumed that c_j does not describe a_i if $i \neq j$, Target probability distribution p is then described as

$$p_a(a_i \mid c_j) := \mathbf{1}_{i=j} \quad \text{and} \quad p_c(c_j \mid a_i) := \mathbf{1}_{i=j}.$$

This is undesirable since the caption that was randomly sampled may, by accident, fully or partially represent the audio recording. The work proposes two-stages pipeline where

Stage 1 trains a model and saves estimated text-audio correspondences on training data using the cross-entropy loss.

Stage 2 is a knowledge distillation procedure. In case of self-distillation, a "student" model is initialized with pre-defined weights from Stage 1 and uses its own correspondences as a target. Better performing setting however used a student model to distill knowledge from three different audio encoders: PaSST, ATST [21], and EfficientAT MobileNetV3 [34]. RoBERTA-large [22] remained the selected model for text embeddings.

Contemporary & Follow-up Contributions

Current research is dedicated exclusively to learning cross-modal similarities. The work by Xie et al. [42] aims to address the lack of relevance annotations raised in [30] in a different way and explores the method that computes on-binary audio-caption relevance scores based on the textual similarities of audio captions. Similarity of captions is measured by cosine similarity of their embeddings. These similarities are then transformed into audio-caption relevance scores using a logistic function. Relevance scores are translated into probability of rating audio samples for a specific textual query using a listwise ranking objective. On a high level, the learning objective is a cross-entropy loss that takes into account both model-predicted relevances and computed relevances based on captions, ensuring that model can predict text-based relevance scores. Their experimental setup for the most part replicates one from [30] except for the new loss function, focusing just on stage 1 training. The results demonstrated that the proposed method outperformed the binary InfoNCE approach retrieval on AudioCaps and Clotho.

DCASE 2025

Leading Submission

Building upon the top-ranked DCASE 2024 Task 8 system [30], the novel approach adds a cluster-based classification task. In preprocessing step, semantically similar captions are grouped into clusters using topic modeling. The model architecture is extended by adding classification heads to both audio and text encoders to predict the cluster label of a caption, which encourages audio encoder to align learned representations with the captions clusters. The total loss combines contrastive loss, distillation loss, and classification loss.

Chapter 3

Experiments

The entire experimental setup builds upon the code of Primus et al. [30].

3.1 Text-Based Audio Retrieval

Language-based audio retrieval systems typically consist of two modality-specific encoders, which learn to map caption and audio samples into a shared embedding space. Feature vectors of both modalities are then normalized and their dot product forms a similarity matrix, with the diagonal elements to be maximized. The general algorithm proceeds as follows (some parts and notation adopted from [8])

1. Let $X_a \in \mathbb{R}^{N \times F \times T}$ represent an audio where F is the number of spectral components such as Mel bins and T is the number of time bins.
2. Let $X_t \in \mathbb{R}^{N \times \ell}$ represent a text where ℓ represents a sequence length.
3. Each audio-text pair is represented in a batch of size N as $\{X_a, X_t\}$.
4. Let $f_a(.)$ be an audio encoder and $f_t(.)$ be a text encoder
5. A batch of size N becomes $\{\hat{X}_a = f_a(X_a), \hat{X}_t = f_t(X_t)\}$ where $\hat{X}_a \in \mathbb{R}^{N \times V}$ are the audio representations of dimensionality V, and $\hat{X}_t \in \mathbb{R}^{N \times U}$ are the text representations of dimensionality U.
6. To project the features into a comparable joint space, linear projections $\mathcal{A} = L_a(X_a)$ and $\mathcal{T} = L_t(X_t)$ are learned, where $\mathcal{A} \in \mathbb{R}^{N \times d}$, $\mathcal{T} \in \mathbb{R}^{N \times d}$, L_a and L_t are the linear projections for audio and text respectively.
7. Each audio feature vector is normalized with L2 norm:

$$\mathcal{A}_i = \frac{\mathbf{A}_i}{\|\mathbf{A}_i\|_2} = \frac{\mathbf{A}_i}{\sqrt{\sum_{j=1}^{d} \mathcal{A}_{i,j}^2}}, \quad i = 1, \ldots, N$$

   Each text feature vector is normalized with L2 norm:

$$\mathcal{T}_i = \frac{\mathbf{T}_i}{\|\mathbf{T}_i\|_2} = \frac{\mathbf{T}_i}{\sqrt{\sum_{j=1}^{d} \mathcal{T}_{i,j}^2}}, \quad i = 1, \ldots, N$$

8. Similarity matrix is computed by

$$C = \mathcal{T} \cdot \mathcal{A}^\top$$

   The matrix has N correct pairs on the diagonal and $N^2 - N$ incorrect pairs off the diagonal

9. Compute loss (NT-Xent or any other choice)

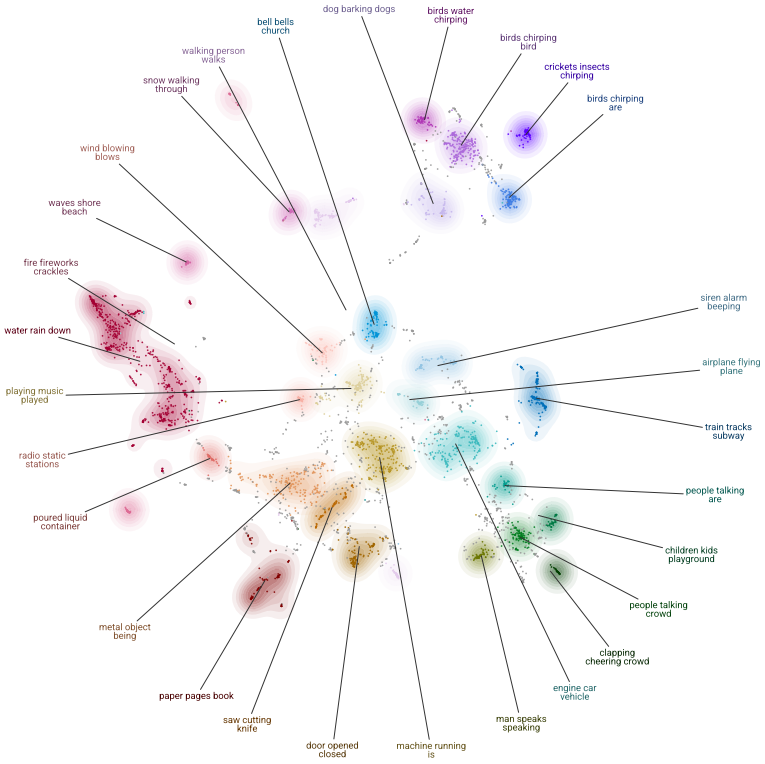


Figure 3.1: Top-25 topics of Clotho with BERTopic and UMAP

3.2 Dataset

Only ClothoV2 dataset is used in experimental setup. Due to limited time and large size of WavCaps, no additional datasets were used for either pretraining or unifying with chosen dataset. The only training data are 3840 audio samles with 5 captions each from Clotho. 1045 audio samples with 5 captions each are used for validation to keep track of learning process and possibility of overfitting. Finally, the test split, equal in size to the validation split, is held out for final evaluation.

For demonstrative purposes, BERTopic [12] is used to process all captions in training set and find corresponding topics. Overall, 78 topics are found. For visibility, top 25 topics are visualized on Figure 3.1 using UMAP [23] to group captions into semantically similar clusters.

3.3 Audio embedding

Audio embedding model for this experimental setup is ATST for reasons of higher temporal resolution. PaSST is also used for experiments with new loss function.

3.3.1 PaSST

In Vision Transformer, an image is divided into small patches, each patch flattened and projected to a vector, and then a Transformer processes the sequence of patch embeddings. Audio Spectrogram Transformer applies the same concept to audio spectrograms

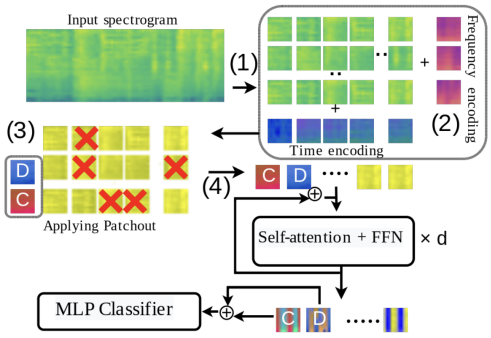


Figure 3.2: PaSST architecture (from [19]) where (1) is the patch extraction and linear projection, (2) frequency and time positional encodings are added, (3) patchout is applied and classification token is added, (4) sequence is flattened and passed through $\mathtt{d}$ layers blocks of self-attention. In final step, a classifier operates on the mean of the transformed C and D tokens.

— the time–frequency representation of an audio signal is treated like an image. The spectrogram is split into time–frequency patches, each patch is linearly projected to an embedding, and the Transformer processes the sequence to model both temporal and spectral dependencies.

While performing better than state-of-the art CNNs for audio tasks, transformers are known for their computational complexity which increases quadratically. *Patchout faSt Spectrogram Transformer* (PaSST) is designed to be efficient yet effective by introducing *patchout* and can be trained on a single consumer-grade GPU. Patchout significantly reduces the computation and memory complexity of training transformers and additionally acts as a regulizer. Furthermore, PaSST enables two positional encodings: one representing the frequency, and one for time. Figure 3.2 summarizes the architecture. PaSST is pre-trained to predict 527 possible classes of AudioSet2M.

For experiments, PaSST encoder was configured with a frequency stride of 10 and a time stride of 10, applied to input spectrograms of size 128 frequency bins × 998 time frames. This configuration results in non-overlapping patches. No patchout augmentation was used. It has a positional encoding for inputs of up to 10 seconds. Depending on chosen segment length, audio is cut in $\frac{30}{\mathtt{l}}$ snippets, where $\mathtt{l}$ is segment length in seconds. The obtained embeddings are then averaged. This encoder is used in baseline runs as it has shown to perform slightly better than ATST and EfficientAT MobileNetV3 in previous work.

3.3.2 ATST

ATST a is transformer-based teacher-student self-supervised learning model. It was also pretrained on AudioSet2M. Unlike PaSST, which was pretrained supervised to predict labels, ATST pretraining involved semi-supervised learning. That is, the model learns a more general audio embedding space, by capturing similarities between audio clips without being trained to predict classes. This also means that no negative examples are used. Given one augmented view of an audio clip, the student network is trained to predict a data representation being identical to the teacher network's prediction on one another augmented view of the same audio clip.

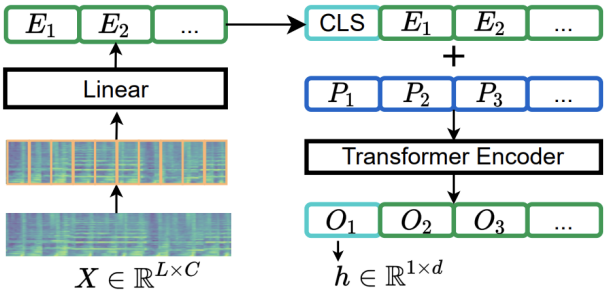


Figure 3.3: The ATST audio encoder takes a mel-spectrogram as input. To reduce sequence length, every four consecutive frames are stacked and projected into a d-dimensional space. A learnable class token is prepended, and trainable positional embeddings are added. the sequence is then processed by a standard Transformer encoder, and the final embedding is obtained from the class token output.

The encoding procedure is illustrated in Figure 3.3. In used setup, the pre-trained transformer encoder of teacher network is used as the feature extractor, while the projector is removed. ATST has positional encoding that is limited to 10 seconds. Obtained embeddings of audio segments are averaged to a single embedding vector. A series of experiments was conducted which shows that the lowering learning rate of this encoder relative to the global learning rate improves performance in given setup.

3.4 Text embedding

For text embedding, RoBERTa Large is used in all experimental runs.

3.4.1 BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers is designed to challenge the problem of uni-directionality of standard language models. BERT uses amasked language model, which masks some of the tokens of the input. During pre-training, the objective is to predict the original vocabulary id of the masked word based only on its context. Such an architecture makes it possible to to pretrain a deep bidirectional Transformer (based on the original Transformer [40]), allows the model to see context on both the right and the left. The first token of an input sequence is a special classification `[CLS]` token. It serves to represent the meaning of the entire sentence and has a purpose of being a sentence-level representation for classification. Another special token `[SEP]` is added to separate sentences. This is visualized on Figure 3.4. The pretraining phase featured *MaskedLM*-masking some percentage of the input tokens at random and replacing them with `[MASK]` token. In addition, model it taught to know relationship between sentences by predicting if given sentence appears next in the corpus. This step is called *Next Sentence Prediction*. Notably, the same architectures are used in both pre-training and fine-tuning.

3.4.2 RoBERTa

A Robustly Optimized BERT Pretraining Approach (*RoBERTa*) has the same underlying architecture as BERT. RoBERTa underwent self-supervised pretraining on the Book-

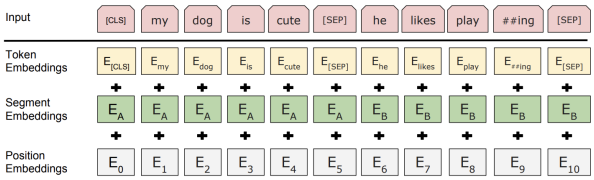


Figure 3.4: BERT input representation(from [5]). Representation is constructed by summing the token, segment, and position embeddings.

Corpus [48] and WikiTexts [26] datasets with better masking strategy(dynamic), longer training and no Next Sentence Prediction objective.

3.5 Loss functions

The ultimate goal of the following loss functions is to maximize all on-diagonal elements all of a similarity matrix s_{ij} to be maximized, while all off-diagonal elements are minimized, trying to converge to an identity matrix of size N for every batch.

3.5.1 InfoNCE & NT-Xent

The InfoNCE loss is calculated as follows:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{N}\exp(s_{ij}/\tau)} \tag{3.1}$$

where τ is a temperature hyperparameter, controlling smoothness of a softmax output.

NT-Xent loss is a specific formulation of the InfoNCE loss which is computed as:

$$\mathcal{L}_{\text{NT-Xent}} = -\frac{1}{2N}\left(\sum_{i=1}^{N}\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{B}\exp(s_{ij}/\tau)} + \sum_{i=1}^{N}\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{N}\exp(s_{ji}/\tau)}\right) \tag{3.2}$$

and is used in the baseline model following [30].

3.5.2 Sigmoid Loss

The Sigmoid Loss is defined as:

$$-\frac{1}{|N|}\sum_{i=1}^{|N|}\sum_{j=1}^{|N|}\log\frac{1}{1+e^{z_{ij}\left(-t\,x_i\cdot y_j+b\right)}} \tag{3.3}$$

where z_{ij} is the label for a given image and text input, which equals 1 if they are paired and -1 otherwise. To mitigate the issue of many negatives dominating the loss, two learnable parameters t (temperature) and b (bias) are introduced.

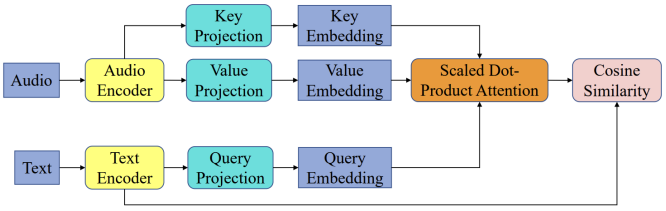


Figure 3.5: Text-Aware Attention Pooling Mechanism (from [43])

3.6 Text-Aware Attention Pooling

To enable the scaled product of each audio and feature vector, obtained audio segments need to be aggregated. A common way is mean-pooling, which is used in the baseline and is simply averaging over feature vectors of corresponding segments. Text-Aware Attention Pooling [43] (TAP) allows a model to reason about the most relevant frames given the text, potentially encoding less redundant information.

3.6.1 Method formulation

Given a text embedding $c_t \in \mathbb{R}^{1\times D}$ and an audio embedding $c_a^{t\times D}$ obtained from text and audio encoders, respectively. t denotes a number of audio frames and D is a size of shared representation. c_t is projected into a query $Q_t \in \mathbb{R}^{1\times D_p}$ and c_a is projected into key $K_a \in \mathbb{R}^{t\times D_p}$ and $V_a \in \mathbb{R}^{t\times D_p}$ matrices, where D_p is chosen projection size. Let W_q, W_k and W_v be learnable matrices in $\mathbb{R}^{D\times D_p}$. Query, key, and value matrices are then computed as:

$$Q_t = LN(c_t^\top)W_q$$
$$K_a = LN(c_a^\top)W_k$$
$$V_a = LN(c_a^\top)W_a$$

where LN denotes a layer normalization layer. Scaled dot-product attention is then applied from text to each relevant frame:

$$\text{Attention}(Q_t, K_a, V_a) = \text{softmax}\left(\frac{Q_t K_a^T}{\sqrt{D_p}}\right)V_a$$

It is only left to embed audio to the same representation size as text $\mathbb{R}^{1\times D}$ by introducing weight matrix $W_o \in \mathbb{R}^{D_p\times D}$ to compute:

$$z_{a|t} = LN\left(\text{Attention}(Q_t, K_a, V_a)W_O\right)$$

where $z_{a|t}$ is audio embedding a depending on text t.

3.7 Synthetic Captions

Another experiment involves training a model with synthetic captions generated with phi-4 [1], a large language model with a relatively small number of parameters (14B) which is run locally. The goal is to double training data by prompting the LLM to generate 5 new captions in addition to existing ones for each audio. This makes up totally $3840 \times 10 = 38400$ captions. The prompt is constructed as follows:

I will provide five captions that describe an audio as well as associated tags in JSON format. Generate exactly five new descriptions with at most 20 words each. Make sure to preserve semantic consistency. Do not add events that are not present. Do not add any numbers. Do not add any names.

3.8 Evaluation metrics

The ground truth caption relevance to an audio is binary which. Mean Average Precision at top-k results (mAP@k) is a rank-aware metric which measures how well the relevant audio samples are ranked near the top. It measures the mean of average precisions (AP) over all the queries. The AP for a query is calculated by averaging the precisions at positions, where relevant items are in the retrieved rank list. It takes into account not only the presence of relevant items, but also their order. AP is thus larger if relevant items are located closer to the top.

Precision at rank k is defined as:

$$P(k) = \frac{\#\text{ relevant items in top-}k}{k}$$

Average precision among 10 items for single query is defined as:

$$AP@10(q) = \frac{1}{R_q} \sum_{k=1}^{10} P(k) \cdot \mathbf{1}\{\text{item at rank } k \text{ is relevant}\}$$

where q is a query and R_q is a number of relevant items. Mean Average Precision among the top-10 results is simply the average over all queries:

$$mAP@10 = \frac{1}{|Q|} \sum_{q \in Q} AP@10(q)$$

where Q is a number of queries

Another metric reported along with mAP@10 is R@k with $k \in \{1, 5, 10\}$. It measures the proportion of correctly identified relevant items in top-K out of total number of relevant items in dataset. It can be defined as

$$R(k) = \frac{\#\text{ relevant items in top-}k}{R_q}$$

3.9 General training procedure and optimization

The training procedure closely follows [30], but without knowledge distillation stage. Batch size of 16 was used throughout all experiments during training, validation and test steps. AdamW was chosen as the optimizer with default PyTorch's default betas $= (0.9, 0.99)$, with one warm-up epoch. The global learning rate is decayed from 2×10^{-5} to 10^{-7} using a cosine schedule. The temperature hyperparameter $\tau = 0.05$ is fixed as the scaling factor. Learnable parameters in Sigmoid Loss are initialized with $t = 1$ and $b = -10$, with a high learning rate starting at 0.1. The shared representation space is $D = 1024$ with projection size of TAP module set to $D_p = 2048$, where a small dropout(0.05) is applied on scaled dot product result. The training process lasts 20 epochs, where mAP@10 is the main evaluation criterion for hyperparameter selection. R@1, R@5 and R@10 are also reported. All runs were conducted on a single NVIDIA RTX 4090 GPU.

Chapter 4

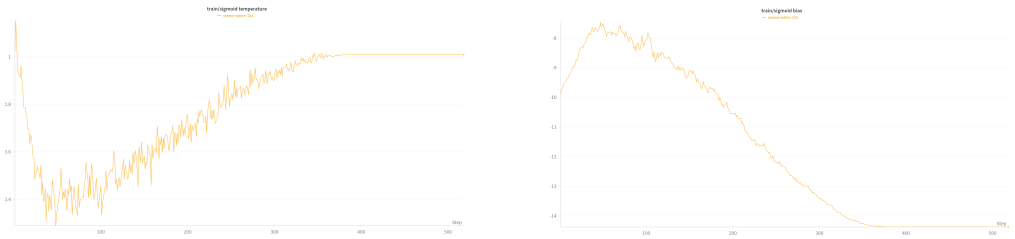
Results

4.1 Sigmoid and NT-Xent Losses

The performance comparison of two functions is shown in Table 4.1. Each experiment was run once using PaSST for audio embedding. Results show that NT-Xent performed slightly better and was chosen in other experiments. NT-Xent remained the loss in all subsequent experiments. Convergence of $\mathtt{t}$ and $\mathtt{b}$ is shown in Figure 4.1.

Loss	R@1	R@5	R@10	mAP@10
NT-Xent	0.1896	0.4493	0.5944	0.3016
Sigmoid Loss	0.1860	0.4549	0.5917	0.2999

Table 4.1: Losses performance comparison



(a) Temperature convergence

(b) Bias convergence

Figure 4.1: Convergence of $\mathtt{t}$ and $\mathtt{b}$ during training

4.2 ATST learning rate

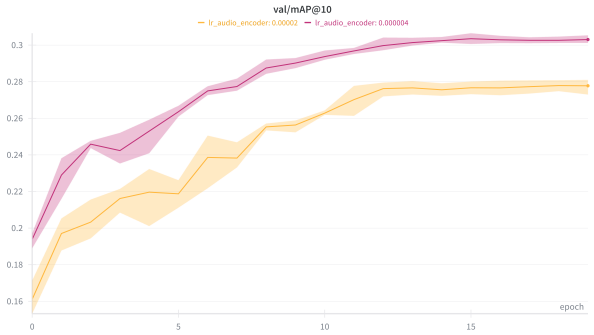
Six runs, with three per learning rate configuration, showed that lowering the ATST learning rate from 2×10^{-5} to 4×10^{-6} led to an improvement on all metrics. Test performance is demonstrated in Table 4.2, while mAP@10 and loss on validation set are shown on Figure 4.2. The absence of improvement in validation loss may indicate that the model did not improve at optimizing the objective.

4.3 Pooling functions evaluation

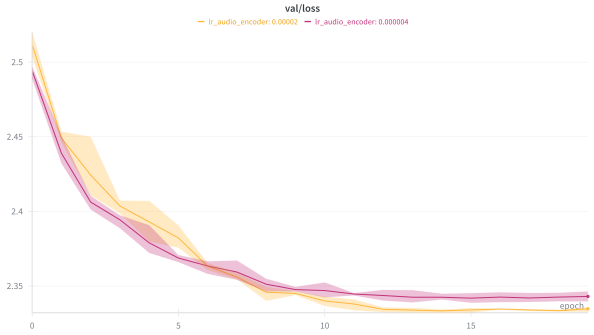
Performance comparison between mean pooling and TAP using ATST is shown in Table 4.3. The best performing configuration from learning rate experiments was chosen as

Metric	ATST: LR=2×10^{-5}	ATST: LR=4×10^{-6}
R@10	0.5614 ± 0.006	**0.59528 ± 0.001**
R@5	0.42144 ± 0.005	**0.45455 ± 0.004**
R@1	0.16797 ± 0.006	**0.18705 ± 0.004**
mAP@10	0.27651 ± 0.003	**0.30069 ± 0.002**

Table 4.2: Average retrieval performance comparison of ATST with different learning rates on the test set. ± denotes standard deviation over 3 runs.



(a) Validation mAP@10 over training epochs



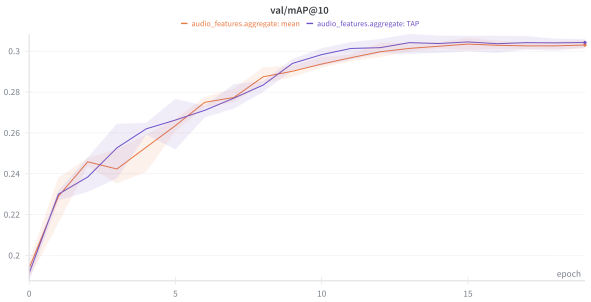
(b) Validation loss over training epochs

Figure 4.2: mAP@10 and validation loss using ATST. Runs are grouped by learning rate of the audio encoder used (orange for LR=2×10^{-5} and pink for LR=4×10^{-6}) Borders along the y-axis denote the maximum and minimum values observed in the experiment.

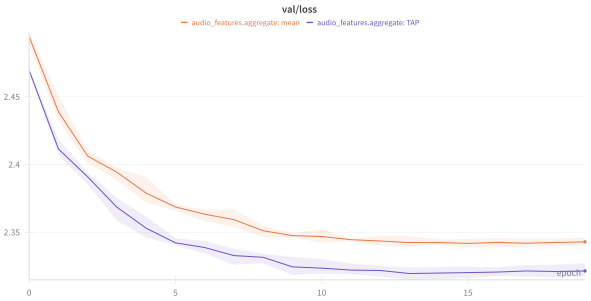
the baseline. mAP@10 and loss on the validation set are shown in Figure 4.3(a). While validation mAP@10 shows no improvement, validation loss indicates better text–audio alignment during training. On the test set, the TAP-enabled configuration slightly outperformed others across metrics, with the best run achieving mAP@10 of 0.31041. Additional experiments with more seeds would be required to confirm statistical significance. At the very least, it is robust and shows no signs of performance drop.

Metric	ATST+Mean	ATST+TAP
R@10	0.59528 ± 0.001	**0.60402 ± 0.009**
R@5	0.45455 ± 0.004	**0.46207 ± 0.009**
R@1	0.18705 ± 0.004	**0.191 ± 0.006**
mAP@10	0.30069 ± 0.002	**0.30722 ± 0.002**

Table 4.3: Average retrieval performance comparison between TAP and mean pooling with ATST on the test set. ± denotes standard deviation over 3 runs.



(a) Validation mAP@10 over training epochs

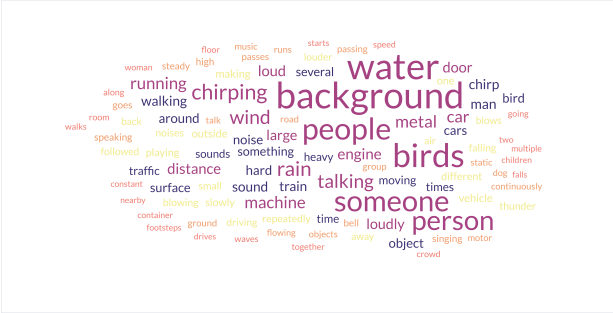


(b) Validation loss over training epochs

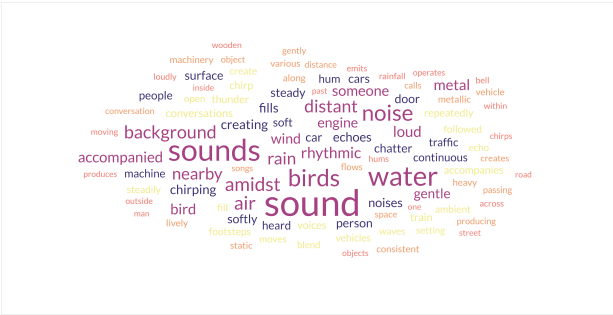
Figure 4.3: mAP@10 and validation loss over training epochs using ATST. Runs are grouped by pooling function used (orange for mean pooling and purple for TAP). Borders along the y-axis denote the maximum and minimum values observed in the experiment.

4.4 Comparison Between Original and Augmented Captions

After doubling training dataset size, performance did not improve in addition to longer training. This could be due to overfitting on synthetic captions and because Clotho is already diverse enough.



(a) Wordcloud: ClothoV2 training data



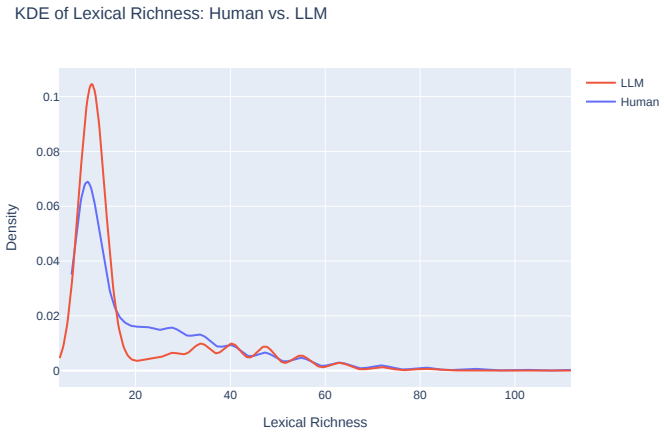
(b) Wordcloud: LLM generated captions

Figure 4.4: Wordclouds of top 100 words of original and LLM captions (Source used: *freewordcloudgenerator.com*)

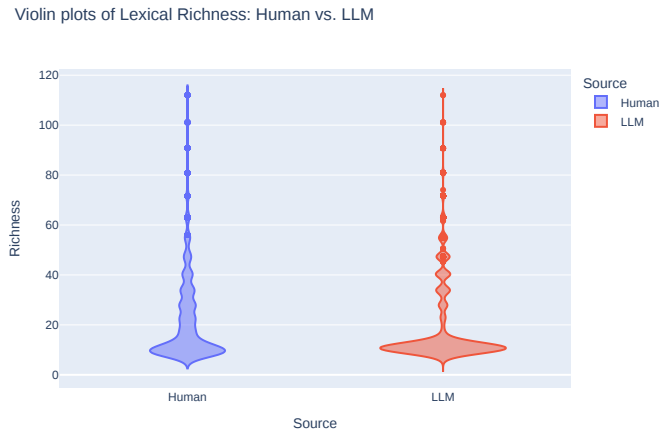
To illustrate difference between captions, Figure 4.4 shows a word cloud of the most frequently used words. Comparing the word clouds, the original captions rely heavily on frequent, object-centric words such as *person*, *water*, and *birds*, whereas LLM captions demonstrate a richer and more descriptive vocabulary, including words like *rhythmic*, *gentle*, and *amidst*. This suggests that LLM-generated captions capture more acoustic and contextual attributes, though at the risk of introducing subjective or hallucinatory descriptions.

Measure of Textual Lexical Diversity [36] computes mean length of sequential words in a text that maintains a minimum threshold Type Token Ratio score. Figure 4.5 shows KDE and Violin Plot and project lexical richness relative to its density in dataset. KDE plot suggests that humans produce a wider spread of lexical richness in the range (10–40), while LLM captions often collapse to very similar low values with a few outliers. Violin plot exhibits similar behavior with lexical richness of humans being more widespread, while the violin is tighter at the bottom for LLM captions.

Human captions have therefore more consistent and broadly distributed lexical richness, while LLMs occasionally generate more verbose outliers. Result can of course vary depending on prompt and chosen LLM.



(a) Lexical richness and density: KDE plot



(b) Lexicall richness and density: Violin plot

Figure 4.5: Lexical richness views for original vs. LLM captions.

Chapter 5

Conclusion

This work focused on the text-to-audio retrieval task using the ClothoV2 dataset and investigated the effectiveness of the Sigmoid Loss, showing that it does not outperform the widely used NT-Xent when paired with PaSST. A more extensive study was conducted on the learning rate hyperparameter, from which ATST substantially benefited. Study of hyperparameter effects should therefore not be neglected in future research attempts. Future work could involve experimenting with the dimensionality of the shared representation space, optimization schedules, or regularization strategies. An architectural change in the form of TAP module also proved to be robust when paired with ATST. Despite modest gains, it still contributes to improved learning. In summary, TAP yielded modest but consistent test improvements without validation gains, suggesting that it refines the learned embedding space rather than drastically altering ranking performance. Future work could explore experimenting with other frame-level audio encoders which TAP could take advantage of. Finally, a visual analysis of captions shows that LLM generated captions are generally lexically less diverse than original Clotho captions, while containing outliers in form of more specific vocabulary.

To further improve performance and remain competitive, it is necessary to use large-scale datasets such as AudioCaps and WavCaps. Several techniques reviewed in this work could be tested in combination with the best-performing setup, such as separately encoding metadata. It would also be worthwhile to experiment with model ensembles and methods that address the problem of non-binary relevance.

Bibliography

[1] Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. 2024. Phi-4 Technical Report. (2024). arXiv: 2412.08905 [cs.CL]. https://arxiv.org/abs/2412.08905.

[2] Gal Chechik, Eugene Ie, Martin Rehn, Samy Bengio, and Dick Lyon. 2008. Large-scale content-based audio retrieval from text queries. In *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*. Michael S. Lew, Alberto Del Bimbo, and Erwin M. Bakker, (Eds.) ACM, pp. 105–112. DOI: 10.1145/1460096.1460115. https://doi.org/10.1145/1460096.1460115.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning* (*Proceedings of Machine Learning Research*, volume 119). Hal Daumé III and Aarti Singh, (Eds.) PMLR, (July 2020), pp. 1597–1607. https://proceedings.mlr.press/v119/chen20j.html.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Jill Burstein, Christy Doran, and Thamar Solorio, (Eds.) Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), pp. 4171–4186. DOI: 10.18653/v1/N19-1423. https://aclanthology.org/N19-1423/.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. (2021). arXiv: 2010.11929 [cs.CV]. https://arxiv.org/abs/2010.11929.

[7] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: an Audio Captioning Dataset. In (May 2020), pp. 736–740. DOI: 10.1109/ICASSP40776.2020.9052990.

[8] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. CLAP Learning Audio Concepts from Natural Language Supervision. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10095889.

[9] Benjamin Elizalde, Shuayb Zarar, and Bhiksha Raj. 2019. Cross Modal Audio Search and Retrieval with Joint Embeddings Based on Text and Audio. In (May 2019), pp. 4095–4099. DOI: 10.1109/ICASSP.2019.8682632.

[10] Jort Gemmeke, Daniel Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Moore, Manoj Plakal, and Marvin Ritter. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In (March 2017), pp. 776–780. DOI: 10.1109/ICASSP.2017.7952261.

[11] Yuan Gong, Yu-An Chung, and James Glass. 2021. AST: Audio Spectrogram Transformer. (2021). arXiv: 2104.01778 [cs.SD]. https://arxiv.org/abs/2104.01778.

[12] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. (2022). arXiv: 2203.05794 [cs.CL]. https://arxiv.org/abs/2203.05794.

[13] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Journal of Machine Learning Research - Proceedings Track*, 9, (January 2010), 297–304.

[14] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. 2022. Audioclip: Extending Clip to Image, Text and Audio. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 976–980. DOI: 10.1109/ICASSP43922.2022.9747631.

[15] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, (Eds.) Volume 33. Curran Associates, Inc., pp. 18661–18673. https://proceedings.neurips.cc/paper_files/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf.

[16] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. 2019. AudioCaps: Generating Captions for Audios in The Wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Jill Burstein, Christy Doran, and Thamar Solorio, (Eds.) Association for Computational Linguistics, Minneapolis, Minnesota, (June 2019), pp. 119–132. DOI: 10.18653/v1/N19-1011. https://aclanthology.org/N19-1011/.

[17] A. Sophia Koepke, Andreea-Maria Oncescu, João F. Henriques, Zeynep Akata, and Samuel Albanie. 2023. Audio Retrieval With Natural Language Queries: A Benchmark Study. *IEEE Transactions on Multimedia*, 25, 2675–2685. ISSN: 1941-0077. DOI: 10.1109/tmm.2022.3149712. http://dx.doi.org/10.1109/TMM.2022.3149712.

[18] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2880–2894. DOI: 10.1109/TASLP.2020.3030497.

[19] Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. 2022. Efficient Training of Audio Transformers with Patchout. In *Interspeech 2022*. ISCA, (September 2022). DOI: 10.21437/Interspeech.2022-227. http://dx.doi.org/10.21437/Interspeech.2022-227.

[20] Rohit Kundu. 2022. The Beginner's Guide to Contrastive Learning. https://www.v7labs.com/blog/contrastive-learning-guide#1-simclr.

[21] Xian Li, Nian Shao, and Xiaofei Li. 2024. Self-Supervised Audio Teacher-Student Transformer for Both Clip-Level and Frame-Level Tasks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 1336–1351. DOI: 10.1109/TASLP.2024.3352248.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (2019). arXiv: 1907.11692 [cs.CL]. https://arxiv.org/abs/1907.11692.

[23] Leland McInnes, John Healy, and James Melville. 2020. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. (2020). arXiv: 1802.03426 [stat.ML]. https://arxiv.org/abs/1802.03426.

[24] Xinhao Mei, Xubo Liu, Jianyuan Sun, Mark D. Plumbley, and Wenwu Wang. 2022. On Metric Learning for Audio-Text Cross-Modal Retrieval. (2022). arXiv: 2203.15537 [eess.AS]. https://arxiv.org/abs/2203.15537.

[25] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. 2024. WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32, 3339–3354. ISSN: 2329-9304. DOI: 10.1109/taslp.2024.3419446. http://dx.doi.org/10.1109/TASLP.2024.3419446.

[26] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer Sentinel Mixture Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=Byj72udxe.

[27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. (2013). arXiv: 1301.3781 [cs.CL]. https://arxiv.org/abs/1301.3781.

[28] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*. ISCA, (September 2019). DOI: 10.21437/interspeech.2019-2680. http://dx.doi.org/10.21437/Interspeech.2019-2680.

[29] Paul Primus, Khaled Koutini, and Gerhard Widmer. 2023. Advancing Natural-Language Based Audio Retrieval with PaSST and Large Audio-Caption Data Sets. (2023). arXiv: 2308.04258 [eess.AS]. https://arxiv.org/abs/2308.04258.

[30] Paul Primus, Florian Schmid, and Gerhard Widmer. 2024. A KNOWLEDGE DISTILLATION APPROACH TO IMPROVING LANGUAGE-BASED AUDIO RETRIEVAL MODELS. (2024). arXiv: 2408.11641 [eess.AS]. https://arxiv.org/abs/2408.11641.

[31] Paul Primus and Gerhard Widmer. 2024. Fusing Audio and Metadata Embeddings Improves Language-Based Audio Retrieval. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pp. 321–325. DOI: 10.23919/EUSIPCO63174.2024.10715116.

[32] Paul Primus and Gerhard Widmer. 2022. Improving Natural-Language-based Audio Retrieval with Transfer Learning and Audio and Text Augmentations. (2022). arXiv: 2208.11460 [cs.SD]. https://arxiv.org/abs/2208.11460.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning* (*Proceedings of Machine Learning Research*, volume 139). Marina Meila and Tong Zhang, (Eds.) PMLR, pp. 8748–8763. https://proceedings.mlr.press/v139/radford21a.html.

[34] Florian Schmid, Khaled Koutini, and Gerhard Widmer. 2023. Efficient Large-Scale Audio Tagging Via Transformer-to-CNN Knowledge Distillation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096110.

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, (June 2015), pp. 815–823. DOI: 10.1109/cvpr.2015.7298682. http://dx.doi.org/10.1109/CVPR.2015.7298682.

[36] Lucas Shen. 2022. LexicalRichness: A small module to compute textual lexical richness. (2022). DOI: 10.5281/zenodo.6607007. https://github.com/LSYS/lexicalrichness.

[37] Malcolm Slaney. 2002. Semantic-audio retrieval. In volume 4. (June 2002), pp. IV–4108. DOI: 10.1109/ICASSP.2002.5745561.

[38] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark Plumbley. 2015. Detection and Classification of Acoustic Scenes and Events. *IEEE Transactions on Multimedia*, 17, (October 2015), 1733–1746. DOI: 10.1109/TMM.2015.2428998.

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2019. Representation Learning with Contrastive Predictive Coding. (2019). arXiv: 1807.03748 [cs.LG]. https://arxiv.org/abs/1807.03748.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, (Eds.) Volume 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[41] Ho-Hsiang Wu, Oriol Nieto, Juan Pablo Bello, and Justin Salamon. 2023. Audio-Text Models Do Not Yet Leverage Natural Language. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10097117.

[42] Huang Xie, Khazar Khorrami, Okko Räsänen, and Tuomas Virtanen. 2025. Text-Based Audio Retrieval by Learning From Similarities Between Audio Captions. *IEEE Signal Processing Letters*, 32, 221–225. DOI: 10.1109/LSP.2024.3511414.

[43] Yifei Xin, Dongchao Yang, and Yuexian Zou. 2023. Improving Text-Audio Retrieval by Text-Aware Attention Pooling and Prior Matrix Revised Loss. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096972.

[44] David Xu. 2024. *Audiosetmix: Enhancing audio-language datasets with llm-assisted augmentations*. Master's thesis. Princeton University.

[45] Xuenan Xu, Zeyu Xie, Mengyue Wu, and Kai Yu. 2022. The SJTU System for DCASE2022 Challenge Task 6: Audio Captioning with Audio-Text Retrieval Pretraining. Technical report. DCASE2022 Challenge.

[46] Zhiyong Yan, Heinrich Dinkel, Yongqing Wang, Jizhong Liu, Junbo Zhang, Yujun Wang, and Bin Wang. 2024. Bridging Language Gaps in Audio-Text Retrieval. In *Interspeech 2024*, pp. 1675–1679. DOI: 10.21437/Interspeech.2024-420.

[47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid Loss for Language Image Pre-Training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11975–11986.

[48] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning Books and Movies: Towards Story-Like Visual Explanations by Watching Movies and Reading Books. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.