Audio Retrieval with Natural Language Queries

Andreea-Maria Oncescu\textsuperscript{1*} A. Sophia Koepke\textsuperscript{2*} João F. Henriques\textsuperscript{1}
Zeynep Akata\textsuperscript{2} Samuel Albanie\textsuperscript{1}

\textsuperscript{1}VGG, University of Oxford \textsuperscript{2}EML, University of Tübingen

https://www.robots.ox.ac.uk/~vgg/research/audio-retrieval/
onescu@robots.ox.ac.uk, a-sophia.koepke@uni-tuebingen.de

Abstract

We consider the task of retrieving audio using free-form natural language queries. To study this problem, which has received limited attention in the existing literature, we introduce challenging new benchmarks for text-based audio retrieval using text annotations sourced from the \textsc{AudioCaps} and \textsc{Clotho} datasets. We then employ these benchmarks to establish baselines for cross-modal audio retrieval, where we demonstrate the benefits of pre-training on diverse audio tasks. We hope that our benchmarks will inspire further research into cross-modal text-based audio retrieval with free-form text queries.

1. Introduction

The large amounts of data being generated in recent years have prompted a real need to search evergrowing databases. A search query typically comprises natural language (text), which allows expressing virtually any concept. Spanning multiple modalities, different retrieval strategies were developed for content as diverse as text (including webpages and books), images\textsuperscript{[1]}, and videos\textsuperscript{[2, 3]}. Surprisingly, while search engines currently exist for these modalities (e.g. Google, Flickr and YouTube, respectively), unstructured audio is not accessible in the same way. The aim of this paper is to address this gap.

It is important to distinguish content-based retrieval from that based on meta-data, such as the title of a video or song or an audio tag. Meta-data retrieval is feasible for manually-curated databases such as song or movie catalogs. However, content-based retrieval is more important in user-generated data, which often has little structure or meta-data. There are methods to search for audio which matches an audio query\textsuperscript{[4, 5]}, but satisfying the requirement to input an example audio query can be difficult for a human (e.g. making convincing frog sounds is difficult). We, on the other hand, propose a framework which enables the querying of a sound database using detailed free-form natural language descriptions of the desired sound (e.g. “A man talking as music is playing followed by a frog croaking.”). This enables the retrieval of audio data which matches the temporal sequence of events in the query instead of just a single class tag. Furthermore, natural language queries are a familiar user interface widely used in current search engines. Therefore, our proposed audio retrieval with free-form text queries could be a first step towards more natural and flexible audio-only search.

Text-based audio retrieval could also be beneficial for video retrieval. The majority of recent works that address the text-based video retrieval task focuses heavily on the visual and text domains\textsuperscript{[1, 2, 6, 7]}. Since audio and visual information inherently have natural semantic alignment for a significant portion of video data, text-based audio retrieval could also be used for querying video databases by only considering the audio stream of the video data. This would allow for video retrieval in the audio domain at reduced computational cost for cases in which audio and visual information correspond, with applications for low-power IoT devices, such as microphones in natural habitats, of particular interest for conservation and biology. Historical archives with extensive sound collections, such as the BBC Sound Effects Archive, would be easier to search, facilitating historical research and public access. Furthermore, text-based retrieval could enable appealing creative applications, such as automatically finding sounds which correspond to input text. This could be especially useful given the growing popularity of audio podcasts and audiobooks which are often supplemented with (background) sounds that match their content.

We propose two new benchmarks for text-based audio retrieval, based on the \textsc{AudioCaps}\textsuperscript{[8]} and \textsc{Clotho}\textsuperscript{[9]} datasets. The former consists of a subset of 10-second audio clips from the AudioSet dataset\textsuperscript{[10]} with additional human-written audio captions, while the latter contains audio captions for sounds sourced from the Freesound platform\textsuperscript{[11]}, varying between 15 and 30 seconds in duration. In contrast to sound event class labels, audio captions contain detailed information about the sounds. A user searching for a particular sound would usually describe the sound using text similar to an audio caption. \textsc{AudioCaps}\textsuperscript{[8]} and \textsc{Clotho}\textsuperscript{[9]} allow to leverage the matching audio-caption pairs to train text-based audio retrieval frameworks. To establish baselines for this task, we adapt existing video retrieval frameworks for audio retrieval. We employ multiple pre-trained audio expert networks and show that using an ensemble of audio experts improves audio retrieval.

In summary, we make three contributions: (1) we introduce two new benchmarks for free-form text-based audio retrieval—to the best of our knowledge, these represent the first public benchmarks for this task; (2) we provide baseline performances with existing multi-modal video retrieval models that we adapt to text-based audio retrieval; (3) we demonstrate the benefits of combining multiple pre-training datasets.

2. Related Work

Our work relates to several themes in the literature: sound event recognition, audio captioning, audio-based retrieval, text-based video retrieval and text-domain audio retrieval. We discuss each of these next.

Sound event recognition. The task of sound event recognition consists of matching audio information with a text label. Research in this area has been driven by challenges, such as DCASE\textsuperscript{[12, 13]}, and by the collection of sound event datasets. These include TUT Acoustic scenes\textsuperscript{[14]}, CHiME-Home\textsuperscript{[15]}, ESC-50\textsuperscript{[16]}, FSDKaggle\textsuperscript{[17]}, and AudioSet\textsuperscript{[10]}. Of relevance to our approach, a number of prior works have employed deep learning for audio comprehension\textsuperscript{[18, 19, 20, 21, 22]} on the AudioSet dataset. However, instead of considering the task

\* Authors contributed equally.
of audio recognition, we focus on the task of retrieval with natural language queries.

**Audio captioning.** Audio captioning consists of generating a text description for a sound [23]. This requires a more detailed understanding of the sound than simply mapping the sound to a set of labels (sound event recognition). Recently, several audio captioning datasets have been introduced, such as CLOTHO [9] which was used in the DCASE automated audio captioning challenge 2020 [24], Audio Caption [25], and AUDIOCAPS [8]. Multiple works have addressed automatic audio captioning on the AUDIOCAPS dataset [26, 27, 28]. In this work, we use the AUDIOCAPS and CLOTHO datasets for cross-modal retrieval.

**Audio-based retrieval.** Multiple content-based audio retrieval frameworks, in particular query by example methods, leverage the similarity of sound features that represent different aspects of sounds (e.g. pitch, or loudness) [29, 30, 31, 35]. More recently, [4] use a siamese neural network framework to learn to encode semantically similar audio close together in the embedding space. [32] address multimedia event detection using only audio data, while [33] tackle near-duplicate video retrieval by audio retrieval. These are purely audio-based methods that are applied to video datasets, but without using visual information. [34] propose a two-step approach for video retrieval which uses audio (coarse) and visual (fine) information together.

**Text-based video retrieval.** More closely related to our work, a number of methods showed that embedding video and text jointly into a shared space (such that their similarity can be computed efficiently) is an effective approach [1, 3, 2, 6, 35, 7]. One particular trend has been to combine cues from several “experts”—pre-trained models that specialise in different tasks (such as object recognition, action classification etc.) to inform the joint embedding. In this work we propose to adapt two such methods: the Mixture of Embedded Experts method of [2] and the Collaborative Experts model of [6], by repurposing them for the task of audio retrieval (described in more detail in Sec. 3).

**Text-domain audio retrieval.** Methods that retrieve audio by matching associated text, such as meta-data or sound event labels, have the implicit assumption that the text is relevant [36], [37] is an early work which links audio and text representations in hierarchical semantic and acoustic spaces. [38] propose a text-based sound retrieval framework which uses single-word audio tags as queries rather than caption-like natural language. The creative approach of [39] learns to align visual, audio, and text representations to enable cross-modal retrieval. Their framework is trained with captioned images and paired image-sound data (sourced from videos) and evaluated using the soundtrack of captioned videos. More recently, [40] use a siamese network to learn a shared latent text and sound space for cross-modal retrieval. While they use class labels as text labels, we study unconstrained text descriptions as queries.

### 3. Methods, Datasets and Benchmarks

In this section, we first formulate the problem of audio retrieval with natural language queries. Next, we describe two cross-modal embedding methods that we adapt for the task of audio retrieval. Finally, we describe the four datasets used in our experimental study, and the two benchmarks that we propose to evaluate performance on the audio retrieval task.

**Problem formulation.** Given a natural language query (i.e. a written description of an audio event recognition), several audio samples, the objective of text-to-audio (abbreviated to t2a) retrieval is to rank the audio samples according to their similarity to the query. We also consider the converse a2t: task, viz. retrieving text with audio queries.

**Methods.** To tackle the problem of audio retrieval, we propose to learn cross-modal embeddings. Specifically, given a collection of N audio samples with corresponding text descriptions, \( \{a_i, t_i\} : i \in \{1, \ldots, N\} \) we aim to learn embedding functions, \( \psi_a \) and \( \psi_t \), that project each audio sample \( a_i \) and text sample \( t_i \) into a shared space, such that \( \psi_a(a_i) \) and \( \psi_t(t_i) \) are close when the text describes the audio, and far apart otherwise. Writing \( s_{ij} \) for the cosine similarity of the audio embedding \( \psi_a(a_i) \) and the text embedding \( \psi_t(t_j) \), we learn the embedding functions by minimising a contrastive ranking loss [41]:

\[
L = \frac{1}{B} \sum_{i=1}^{B} \left[ m + s_{ij} - s_{ii}\right]_+ + \left[m + s_{ji} - s_{ij}\right]_+ \tag{1}
\]

where \( B \) denotes the batch size, \( m \) the margin (set as a hyperparameter), and \( [x]_+ = \max(\cdot, 0) \) the hinge function.

We consider two frameworks for learning such embedding functions \( \psi_a \) and \( \psi_t \):

1. **Mixture-of-Embedded Experts (MoEE)** [2] and Collaborative-Experts (CE) [6]. MoEE, originally designed for video-text retrieval, constructs its video encoder from a collection of “experts” (features extracted from networks pre-trained for object recognition, action classification, sound classification, etc.) which are computed for each video, aggregated along their temporal dimension with NetVLAD [42], projected to a lower dimension via a self-gated linear map and then L2-normalised. The text encoder first embeds each word token with word2vec [43] and aggregates the results with NetVLAD [42]. The result is projected by a sequence of self-gated linear maps (one for each expert) into a shared embedding space with the outputs of the video encoder. Finally, a scalar-weighted sum of the embedded experts in each joint space is used to compute the overall cosine similarity between the video and text (see [2] for more details). CE adapts the same text encoder as MoEE and similarly makes use of multiple video experts. However, rather than projecting them directly into independent spaces against the embedded text, CE first applies a “collaborative gating” mechanism, which filters each expert with an element-wise attention mask that is generated with a small MLP that ingests all pairwise combinations of experts (see [6] for further details).

To adapt these frameworks for audio retrieval, we use the same text encoder structures \( \psi_t \), and build audio encoders \( \psi_a \) by mimicking the structure of their video encoders, replacing the “video experts” with “audio experts” (described in Sec. 4).

**Datasets.** As the primary focus of our work, we study two audio-centric datasets—these are datasets which comprise audio streams (sometimes with accompanying visual streams) paired with natural language descriptions that focus explicitly on the content of the audio track. To explore differences between audio retrieval and video retrieval, we also consider two visual-centric datasets, that comprise audio and video streams paired with natural language which focus primarily (though not always exclusively) on the content of the video stream. Details of the four datasets we employ are given next.

1. **AUDIOCAPS** [8] (**audio-centric**) is a dataset of sounds with event descriptions that was introduced for the task of audio captioning, with sounds sourced from the AudioSet dataset [10]. Annotators were provided the audio tracks together with category hints (and with additional video hints if needed). We use a subset of the data, excluding a small number of samples for which either: (i) the YouTube-hosted source video is no longer
Table 1: Audio retrieval on audio-centric and visual-centric datasets. Performance is strongest on the audio-centric AUDIOCAPS dataset and weakest on the visual-centric ACTIVITY-NET-CAPTIONS (ACTNETCAPS) dataset.}

| Dataset          | Audio Focus | Pool       | Text  | Audio  | Text  | R31↑ | R10↑ T | R31↑ | R10↑ T |
|------------------|-------------|------------|-------|--------|-------|------|--------|------|--------|
| AUDIOCAPS [3]    | audio       | 815        | 180.10 02/62.01 05 | 21.93 08/62.71 16 | 6.55 08/31.41 16 |
| CLOTHO [9]       | audio       | 1045       | 4.00 02/25.45 03 | 1.84 04/25.64 17 | 1.84 04/16.82 09 |
| ACTNETCAPS [46]  | visual      | 497        | 1.77 01/23.41 01 | 1.77 01/5.86 02 | 8.78 01/10.83 00 |
| QUERYD [17]      | visual      | 1954       | 3.75 02/17.00 11 | 3.80 02/16.20 19 | 3.80 02/16.20 19 |

Table 2: The influence of different experts on AUDIOCAPS. A comparison of audio and visual experts (applied to the video from which the audio was sourced) using CE [6]. We observe that audio features are significantly more effective than visual features (which nevertheless provide some complementary signal as can be seen when jointly using audio and visual features), three different randomly seeded runs.

Implementation details. To encode the audio signal, we use two pre-trained audio feature extractors: VGGish pre-trained for audio classification on YouTube8M [50], and ResNet18 pre-trained for audio classification on the VGGSound dataset [41]. All audio retrieval models were trained with a batch size of 128 using the contrastive ranking loss (Eqn. 1), with \( m \) set to 0.2. The learning rate was set to 0.01 with a weight decay of 0.001. Further details on optimisation, pre-trained experts (for audio and visual modalities) and hyperparameter choices are provided in the supplementary material.

Audio-centric vs. visual-centric queries. We first investigate audio retrieval with audio-centric and visual-centric queries. For this experiment, we use CE with a single expert (VGGish audio features). In Table 1, we observe that performance is strongest on the audio-centric AUDIOCAPS dataset and weakest overall on the ACTIVITY-NET-CAPTIONS dataset. We note that the CLOTHO dataset is particularly challenging, with performance weaker (accounting for pool size) than the visual-centric QUERYD dataset. We hypothesise that this is for two reasons: (1) the significantly smaller training set size of CLOTHO, compared to all other datasets, (2) CLOTHO was constructed by selecting audio samples that were as diverse as possible, making it a potentially more difficult benchmark. We note, however, that the QUERYD experiments suggest that computationally efficient video retrieval using only the audio stream can still be obtained, although at a lower accuracy.

Ablation study. We next conduct an ablation study to investigate the effectiveness of different experts for audio retrieval on the AUDIOCAPS dataset. We present the results in Tab. 2, where we observe that audio experts significantly outperform visual experts (pre-trained models for visual tasks like scene classification, which we compute from the video from which the audio was sourced). We note that the combination of audio and visual experts performs strongest overall, suggesting that the audio-centric queries contain information that is more accessible from the visual modality. The strongest audio-only retrieval is achieved by combining VGGish and VGGSound features—

We therefore adopt this setting for the remaining experiments.

Additionally, we experimented with using speech features (word2vec [43] encodings of speech-to-text transcriptions) [51] of the audio stream). However, this did not improve the re-

---

1We make the sample list publicly available at the project page [45].

2Since the AudioCaps test set is a subset of the AudioSet training set (unbalanced), we do not use audio experts pre-trained on AudioSet.
performance after pre-training models for retrieval on AudioCaps using CE with VGGish and VGGSound features. For an input text query, we visualise the top 3 retrieved audio samples using a video frame from the corresponding videos (the audio can be heard at the project webpage [45]). We mark the audio samples which correspond to the query with green boxes. Successful retrievals are shown in a), failures in b). Note, in particular, the examples in b), where the model’s top 1 retrieved audio is not the correct one, but the retrieved results nevertheless sound reasonable (visually convincing results are marked with yellow boxes).

Table 3: Audio retrieval benchmarks. Text-audio and audio-text retrieval results for MoEE [2] and CE [6] with VGGish and VGGSound features on the proposed AudioCaps and CLOTHO retrieval benchmarks. Pre-training on AudioCaps improves the performance on CLOTHO.

6. Conclusion
We introduced a new benchmark for natural language based audio retrieval, and provided baseline results by adapting strong multi-modal video retrieval methods. Our results show that these methods are relatively well-suited for the audio retrieval task, however there is room for improvement, as expected for an under-explored problem. We hope that our proposed benchmarks will facilitate the development of future audio search engines, and make this large fraction of the world’s produced media available for public use.

Acknowledgements. AMO was supported by an EPSRC DTaS Studentship. ASK and ZA were supported by the ERC (853489 - DEXIM) and by the DFG (2064/1 – Project number 390727645). JFH is supported by the Royal Academy of Engineering (RF\’2018\’18\’163). SA was supported by EPSRC EP/T028572/1 Visual AI.

References
[1] J. Dong et al., “Word2visualce: Image and video to sentence matching by visual feature prediction,” arXiv:1604.06838, 2016.
[2] A. Miech et al., “Learning a text-video embedding from incomplete and heterogeneous data,” arXiv:1804.02516, 2018.
[3] N. C. Mithun et al., “Learning joint embedding with multimodal cues for cross-modal video-text retrieval,” in Proc. ACM ICMR, 2018.
[4] P. Manocha et al., “Content-based representations of audio using siamese neural networks,” in Proc. ICASSP, 2018.
[5] I. Lallemand et al., “Content-based retrieval of environmental sounds by multiresolution analysis,” in Proc. SMC, 2012.
[6] Y. Liu et al., “Use what you have: Video retrieval using representations from collaborative experts,” in Proc. BMVC, 2019.

Figure 1: Qualitative results. Text-based audio retrieval results on AudioCaps using CE with VGGish and VGGSound features. For an input text query, we visualise the top 3 retrieved audio samples using a video frame from the corresponding videos (the audio can be heard at the project webpage [45]). We mark the audio samples which correspond to the query with green boxes. Successful retrievals are shown in a), failures in b). Note, in particular, the examples in b), where the model’s top 1 retrieved audio is not the correct one, but the retrieved results nevertheless sound reasonable (visually convincing results are marked with yellow boxes).

Figure 2: The influence of training data scale. CE retrieval performance on AudioCaps with different proportions of the available AudioCaps training data.
A. Supplementary Material

We give details on the pre-trained experts used, and on optimisation and hyperparameter choices below.

A.1. Pre-trained experts

This section contains information about the audio and visual expert features that were obtained from pre-trained networks.

A.1.1. Audio experts

We use two different audio experts, which we abbreviate as VGGish and VGGSound.

VGGish. These audio features are obtained with a VGGish model [50], trained for audio classification on the YouTube-8M dataset [53]. To produce the input for this model, the audio stream of each video is re-sampled to a 16kHz mono signal, converted to an STFT with a window size of 25ms and a hop size of 10ms with a Hann window, then mapped to a 64 bin log mel spectrogram. Finally, the features are parsed into non-overlapping 0.96s collections of frames (each collection comprises 96 frames, each of 10ms duration), which is mapped to a 128-dimensional feature vector.

VGGSound. These features are extracted using a ResNet-18 model [54] that has been pre-trained on the VGGSound dataset (model H) [44]. We modify the last average pooling layer to aggregate along the frequency dimension, but keep the full temporal dimension. This results in features of dimension t x 512, where t denotes the number of time steps.

A.1.2. Visual experts

We use three different visual experts, abbreviated as Inst, Scene, and R2P1D.

Inst. These features are extracted using a ResNeXt-101 model [55] that has been pre-trained on Instagram hashtags [56] and finetuned on ImageNet [57] for the task of image classification. Features are extracted from frames extracted at 25 fps, where each frame is resized to 224 × 224 pixels. Embeddings are 2048-dimensional.

Scene. Scene features are extracted from 224×224 pixel centre crops with a DenseNet-161 model [58] pre-trained on Places365 [59]. Embeddings are 2208-dimensional.

R2P1D. Features are extracted with a 34-layer R(2+1)D model [60] trained on IG-65M [61] which processes clips of 8 consecutive 112 × 112 pixel frames, extracted at 30 fps. Embeddings are 512-dimensional.

A.2. Optimisation details and hyperparameter choices

A.2.1. Optimiser

We used the Lookahead solver [62] in combination with RAdam [63] (implementation by [64]) with an initial learning rate of 0.01 and weight decay of 0.001. We use a learning rate decay for each parameter group with a factor of 0.95 every epoch. Most models were trained for a maximum of 20 epochs whilst the Activity-Net model was trained for 40 epochs. We chose the models that gave the best performance on the geometric mean of R@1, R@5, and R@10.

A.2.2. NetVLAD hyperparameters

For text, we used 20 VLAD clusters and one ghost cluster. For the VGGish and VGGSound features we used 16 VLAD clusters.

A.2.3. Zero-padding hyperparameters

In the main paper, we used zero-padding of variable length experts to a common shape $z = (z_t, z_a, z_v)$. For AudioCAPs we have used $z_t = 20$ for text tokens, $z_a = 29$ for VGGish feature tokens, and $z_v = 29$ for the VGGSound features. For CLOTHO we have used zero-padding with $z_t = 21$ for text tokens, $z_a = 31$ for VGGish feature tokens, and $z_v = 95$ for the VGGSound features.

For QuerYD we have used $z_t = 70$ for text tokens and $z_a = 500$ for VGGish feature tokens. Lastly, for Activity-Net we used $z_t = 20$ for text tokens and $z_a = 29$ for VGGish feature tokens.