# TAKE IT FOR GRANTED: IMPROVING LANGUAGE-BASED AUDIO RETRIEVAL WITH LARGE LANGUAGE MODELS

Technical Report

*Jan Kulik*, Bartłomiej Zgórzyński*, Juliusz Kruk, Ivan Ryzhankow, Anna Plęs, Theodore Lamort de Gail*

Samsung R&D Institute Poland, Warsaw, Poland
{j.kulik, b.zgorzynski, j.kruk, i.ryzhankow, a.ples, t.lamort}@samsung.com

## ABSTRACT

In this report, we present our solution to DCASE 2024 task 8: Language-Based Audio Retrieval. We employ a bi-encoder architecture trained using InfoNCE loss. The audio encoder is a pre-trained PaSST-S model, while the text encoder is either a pre-trained GTE-large or RoBERTa-large model.

In order to increase the amount of training data, we obtain 10.8 million video-caption pairs from various open-source datasets. We then extract useful audio-caption pairs and evaluate them using our model to filter out low-quality samples. Finally, we use GPT-4o to rephrase the video captions to make them more audio-oriented. In addition, we use GPT-4o for back-translation and GPT-3.5-turbo for Clotho caption mixing.

We achieve 43.69% mAP@10 on the development-testing split of Clotho using an ensemble solution, and 40.78% mAP@10 with a single model.

***Index Terms—*** Language-Based Audio Retrieval, DCASE 2024, Bi-encoder architecture, InfoNCE Loss, Multimodal learning

## 1. INTRODUCTION

Language-Based Audio Retrieval is the task of retrieving audio samples from a database based on a natural language description. Our system employs a standard approach, where text and audio encoders generate text and audio embeddings. These embeddings are then compared using cosine similarity to find the most relevant audio samples that match given description.

In order to build an efficient model, it was necessary to use state-of-the-art text and audio encoders together with the involvement of large amount of high-quality data for training. The primary inspiration for the architecture was a DCASE 2023 submission by Paul Primus [1], which employed PaSST [2], one of the leading audio models. Furthermore, of the many text encoders we tried, the best results were achieved by GTE-large [3] and RoBERTa-large [4]. Detailed information about the architecture can be found in section 2. We trained this architecture using a dataset comprised of audio-caption pairs. Comprehensive details about the data and the augmentations applied are provided in section 3. An explanation of the training steps is given in section 4, followed by a description of the submission and final metrics in section 5.

*These authors contributed equally to this work

## 2. ARCHITECTURE

To approach the problem of Language-Based Audio Retrieval, we adapt a bi-encoder architecture designed to estimate similarity between audio and text data. Input audio and text are mapped to a 1,024-dimensional latent space, where pairs with similar meanings are positioned close to each other, while pairs with different meanings are positioned further apart. The similarity between embeddings is determined using cosine similarity. For textual embeddings, we utilize either the GTE-large model or the RoBERTa-large model. To encode audio, we use the PaSST-S encoder. We decided to train the entire network simultaneously without freezing any layers.

### 2.1. GTE-large

The GTE-large (General Text Embeddings Large) model, developed by the Institute for Intelligent Computing at Alibaba Group, is a text embedding model. In our architecture, we utilized the 1.5-upgraded version of this model. It can handle a maximum context length of 8,192 tokens, producing embeddings of size 1,024. Trained using contrastive learning, the GTE-large model achieves state-of-the-art performance on benchmarks within its size category. GTE-large together with its projection layer comprises 435 million parameters.

### 2.2. RoBERTa-large

The RoBERTa-large (Robustly Optimized BERT Approach Large) model, developed by Facebook AI, is a text embedding model pre-trained on an extensive corpus of English data. This model supports a maximum context length of 512 tokens and generates text embeddings with 1,024 dimensions. RoBERTa-large is trained on a significantly larger dataset compared to its predecessor, BERT [5], and utilizes dynamic masking to improve generalization. Along with other enhancements, RoBERTa-large outperforms many contemporary models in its category. RoBERTa-large together with its projection layer comprises 355 million parameters.

### 2.3. PaSST-S

The PaSST-S (Patchout faSt Spectrogram Transformer Small) model for advanced audio processing, is a spectrogram-based transformer model developed in 2022 by the Institute of Computational Perception and LIT AI Lab at Johannes Kepler University Linz. The model processes audio spectrograms by splitting them into patches and selectively dropping some during training, a proposed technique named patchout. It aims to enhance generalization, while reducing computation complexity. Trained on extensive audio

datasets, the PaSST-S model achieves state-of-the-art performance on various audio benchmarks. The model generates embeddings with a size of 768. PaSST-S together with its projection layer comprises 87 million parameters.

## 3. DATA AND AUGMENTATIONS

One of the most critical challenges in developing a successful audio-retrieval model is acquiring high-quality training data. In this context, quality encompasses three essential components: clean and diverse audio samples, meaningful and precise captions, and the strength of the connection between the captions and the audio samples. During the development of Language-Based Audio Retrieval we employed three open-source datasets: **Clotho v2.1** [6], **AudioCaps** [7] and **WavCaps** [8]. In addition, to address the limited amount of data, we obtained multiple high-quality open-source video-caption datasets to extract valuable audio-caption pairs and created a custom **VideoCaps** dataset. In total, we gathered over 450,000 audio-caption pairs from the audio datasets and 70,000 audio-caption pairs from the VideoCaps dataset. Details of the datasets are listed below.

### 3.1. Clotho v2.1

Clotho serves as the foundational dataset for this task. The official version comprises 6,974 audio samples and 34,870 captions, with each audio sample being paired with five captions. The dataset we utilized consists of 5,925 audio samples. This includes 3,839 samples in the development split used for training, 1,045 samples in the validation split, and 1,045 samples in the evaluation split. In total, the dataset provided us with 19,195 audio-caption pairs for training and 5,225 pairs for both validation and evaluation.

### 3.2. AudioCaps

AudioCaps is a high-quality audio-caption dataset with human-written captions, collected via crowdsourcing. The dataset we used for training our model consists of 43,698 audio samples with one caption and 1,293 audio samples with 5 captions each. Altogether, the dataset of 50,161 audio-caption pairs was used for training.

### 3.3. WavCaps

WavCaps is a large-scale, weakly-labeled audio-caption dataset. The authors utilized the GPT-3.5-turbo model to process and refine raw captions for audio samples collected from various sources. Consequently, WavCaps stands as the largest open-source audio-caption dataset, comprising over 400,000 audio-caption pairs. However, the quality of the captions is notably lower compared to those found in the Clotho and AudioCaps datasets. The dataset we used for training, without empty samples and those excluded from the challenge, consists of 401,112 audio-caption pairs.

### 3.4. VideoCaps

In order to create a new high-quality dataset, we collected commonly used video-caption datasets: Activity-Net [9], Charades-Ego [10], MSRVTT [11], MSVD [12], VATEX [13], VIOLIN [14] and WebVid [15]. This resulted in obtaining around 10.8 million samples. Then, we extracted samples that contained valid audio, which narrowed the dataset down to around 770,000 audio-caption pairs. The main challenge was that many of the captions were primarily video-focused and did not contain any meaningful information about the audio content. Thus, in order to filter out such cases, an early version of our audio retrieval model trained only on Clotho, AudioCaps and WavCaps was used to calculate cosine similarity of all ground truth audio-caption pairs in the extracted dataset. This approach can serve as an effective method to evaluate the quality of the dataset and remove irrelevant samples. In order to remove low-quality pairs, top 100,000 samples were selected for additional processing. Moreover, top 70,000 samples were selected for further comparisons. Average cosine similarity of the selected subsets is presented in Table 1.

| Dataset subset | Average cosine similarity [-] |
|---|---|
| All data (770,000 samples) | 0.2047 |
| Top 100,000 samples | 0.3544 |
| Top 70,000 samples | 0.3668 |

Table 1: Average cosine similarity of selected data subsets

Even though only the top audio-caption pairs were selected, they still contained significant amount of visual context that would be irrelevant during audio retrieval training. Therefore, it was decided to use Large Language Models (LLMs) to rephrase original captions and improve their quality. The following prompt was used as input:

> *You will be given video captions. Rephrase them and remove parts that couldn't possibly be inferred from audio events. Remove any details from the captions that refer to visual or spoken events. Focus on the audio content only. Remove dates, time and names of places and persons. Do not write introductions or explanations. Each audio caption should be one sentence with less than 15 words. Use grammatical sentences. Your reply should have a JSON format. Make sure that the generated JSON is valid: {"caption 0": "caption 0 here", "caption 1": "caption 1 here"} etc.*

This method was tested on GPT-4o, since it provides a good trade-off between performance and cost. Furthermore, temperature settings of 0.7 and 1.0 were tested in order to assess how it impacts the quality of resulting captions. The results can be seen in Table 2.

| Model | Average cosine similarity [-] |
|---|---|
| GPT-4o, temperature=1.0 | 0.3422 |
| GPT-4o, temperature=0.7 | 0.3435 |

Table 2: Average cosine similarity after processing with GPT-4o

As shown, the temperature setting of 0.7 tends to outperform 1.0. Notably, the rephrased datasets have lower average cosine similarity than the original ones. Upon closer investigation, it was found that some of the rephrased captions were empty or contained irrelevant information, possibly because GPT-4o deemed them as inadequate or did not manage to solve the task properly. Therefore, top 70,000 samples were extracted from the dataset processed with the temperature of 0.7 and their average cosine similarity was calculated to be 0.3735, which outperforms the top 70,000 samples from the non-rephrased dataset (see Table 1). These rephrased top 70,000 samples are then used for training.

### 3.5. Augmentations

For audio augmentations, we used only those integrated into the PaSST-S model, including time and frequency masking and structured patchout. At the same time, we extensively augmented the captions, recognizing that many were of low quality. To achieve this, we leveraged LLMs such as GPT-4o and GPT-3.5-turbo. Utilized text augmentation techniques can be found below:

1. **Random deletion.** A random word from a caption is deleted with a given probability.
2. **Synonym replacement.** A random word from a caption is replaced with a synonym with a given probability, using the NLTK library [16].
3. **Back-translation.** Each caption is translated to a random language and then back to English using GPT-4o. We use this method on the entire training split of Clotho thus obtaining 19,195 augmented samples. The following prompt was used for this task:

   *You will be given audio captions. The captions are going to be used for training of an audio captioning model. Translate every caption to a random language and then translate it back to English. When translating, feel free to make proper adjustments to ensure the phrase is natural and coherent. Do not comment on translations. Your reply should have a JSON format, make sure that the generated JSON is valid (e.g. has proper quotation marks at proper locations): {"caption_random_language": "new caption in random language here", "caption_english": "new caption in english here"}*
4. **LLM mixing.** Waveforms of audio samples from the Clotho dataset are mixed together and GPT-3.5-turbo is prompted to combine the corresponding captions. This process results in the creation of 50,000 new audio-caption pairs. The following prompt was used as input for this task:

   *You will be given a list of audio captions. Your task is to mix them together to generate a new caption. The caption that you generate should be a mix of all the input captions. Keep the generated caption under 15 words. Do not write introductions or explanations. The caption should be a natural and coherent sentence in the style of the input captions. The captions are not chronological, so don't refer to time dependencies between them.*

## 4. TRAINING

To train our system, we employ InfoNCE loss with a trainable temperature. After calculating embeddings of all $n$ audios and texts from a given batch, we compute the similarity matrix $S$, where $S_{ij}$ denotes the similarity between text $i$ and audio $j$. The diagonal of the matrix represents matching pairs, while all other elements are considered non-matching. Then, we calculate the mean cross-entropy loss on each row (text-to-audio loss) and each column (audio-to-text loss) after applying softmax function. The final loss is the mean of the audio-to-text and text-to-audio components.

We analyze 30-second audio segments based on Clotho's maximal audio length. The audio encoder processes 10-second segments, and thus we split the input audio into 10-second windows with a certain hop size, thereby introducing additional overlap between windows. Subsequently, we average all embeddings from a given audio. We opted for a 10-second hop size for models using GTE-large as the text encoder and a 5-second hop size for models using RoBERTa-large.

To update the model's parameters, we choose the AdamW optimizer with a batch size of 128. Additionally, we use a cosine decay learning rate scheduler with warmup. For selecting the best model checkpoints during training, we monitor the mAP@10 value on the validation set, which is conducted twice within each training epoch.

The model training consists of three main steps:

**Initial training** - during this stage, the training utilizes Clotho-training, AudioCaps and WavCaps with Clotho-validation employed for validation purposes. The training consists of 16 epochs, with a learning rate schedule from $1 \times 10^{-5}$ to $5 \times 10^{-7}$. We utilize structured patchout of 15 and 2 for time and frequency dimensions, respectively. Additionally, random deletion and synonym replacement are applied with a probability of 0.8.

Two models were chosen for further fine-tuning: one using GTE-large and the other RoBERTa-large as text encoders. They achieved mAP@10 scores of 37.21 and 37.72, respectively.

**Fine-tuning** - for training, we utilize Clotho-training, AudioCaps, and VideoCaps, with Clotho-validation used for validation purposes. The number of epochs has been reduced to 6, and the learning rate has been decreased to range from $3 \times 10^{-6}$ to $6 \times 10^{-8}$. In addition to initial training data augmentations, we also employ LLM mixing and backtranslation. To increase model regularization, we changed the optimizer weight decay from 0.0 to 0.1. The model with GTE-large achieved an mAP@10 of 39.94, while the model with RoBERTa-large achieved an mAP@10 of 39.83.

**Second fine-tuning** - the trainings of our models demonstrated high stability, leading us to utilize Clotho-validation for training and Clotho-testing for validation. We conducted an extensive grid search across parameters including learning rates, warmup lengths and datasets used to generate multiple checkpoints for ensemble aggregation. The best GTE-large model achieved an mAP@10 of 40.78, while the best RoBERTa-large model achieved an mAP@10 of 40.58.

## 5. SUBMISSION

For the DCASE 2024 task 8: Language-Based Audio Retrieval, we prepared four submissions. For submission 1, we have decided to use our best single model, which uses GTE-large as text encoder. This model consists of approximately 522 million parameters.

Submissions 2 and 3 consist of ensembles of different models utilizing GTE-large and RoBERTa-large text encoders. All models were produced during a secondary fine-tuning process. Our results demonstrate that the combination of GTE-large and RoBERTa-large models yields the most substantial improvement in mAP@10. Conversely, employing multiple instances of the same architecture results in only a marginal increase in performance. Submission 2 utilized an ensemble of 3 models, while Submission 3 employed 8 models.

For submission 4, we applied the Hungarian Algorithm for solving the assignment problem. Our objective was to maximize the overall model confidence across the entire evaluation dataset by ensuring each audio is matched to a caption exactly once. Specifically, we arranged these predictions by assigning the first-found audio file

to prediction 1, and filling predictions 2-10 with the subsequent files selected by the model without replacement. The results achieved by our submitted solutions are presented in Table 3.

| Submission name | R@1 | R@5 | R@10 | mAP@10 |
|---|---|---|---|---|
| SRPOL_1 | 28.71 | 57.38 | 70.87 | 40.78 |
| SRPOL_2 | 30.22 | 59.18 | 73.15 | 42.55 |
| SRPOL_3 | 30.07 | 59.67 | 73.07 | 42.62 |
| SRPOL_4 | 31.37 | 60.08 | 73.30 | 43.69 |

Table 3: Evaluation scores on Clotho v2.1

# References


[1] Paul Primus, Khaled Koutini, and Gerhard Widmer. *Advancing Natural-Language Based Audio Retrieval with PaSST and Large Audio-Caption Data Sets*. 2023. arXiv: 2308.04258.

[2] Khaled Koutini et al. "Efficient Training of Audio Transformers with Patchout". In: *Interspeech 2022*. ISCA, 2022. URL: http://dx.doi.org/10.21437/Interspeech.2022-227.

[3] Zehan Li et al. "Towards general text embeddings with multi-stage contrastive learning". In: *arXiv preprint arXiv:2308.03281* (2023).

[4] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[5] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805.

[6] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. *Clotho: An Audio Captioning Dataset*. 2019. arXiv: 1910.09387.

[7] Chris Dongjoo Kim et al. "AudioCaps: Generating Captions for Audios in The Wild". In: *NAACL-HLT*. 2019.

[8] Xinhao Mei et al. *WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research*. 2023. arXiv: 2303.17395.

[9] Bernard Ghanem Fabian Caba Heilbron Victor Escorcia and Juan Carlos Niebles. "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 961–970.

[10] Gunnar A. Sigurdsson et al. *Charades-Ego: A Large-Scale Dataset of Paired Third and First Person Videos*. 2018. arXiv: 1804.09626.

[11] Ganchao Tan et al. *Learning to Discretely Compose Reasoning Module Networks for Video Captioning*. 2020. arXiv: 2007.09049.

[12] David L. Chen and William B. Dolan. "Collecting Highly Parallel Data for Paraphrase Evaluation". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL-2011)*. Portland, OR, June 2011.

[13] Xin Wang et al. *VATEX: A Large-Scale, High-Quality Multilingual Dataset for Video-and-Language Research*. 2020. arXiv: 1904.03493.

[14] Jingzhou Liu et al. *VIOLIN: A Large-Scale Dataset for Video-and-Language Inference*. 2020. arXiv: 2003.11618.

[15] Max Bain et al. "Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval". In: *IEEE International Conference on Computer Vision*. 2021.

[16] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.