TRAINING STRATEGY OF MASSIVE TEXT-TO-AUDIO MODELS AND GPT-BASED QUERY-AUGMENTATION

Technical Report

Hokuto Munakata, Taichi Nishimura, Shota Nakada, Tatsuya Komatsu

LY Corporation, Japan

ABSTRACT

This report describes our system submitted to the DCASE 2024 Task 8: Language-based Audio Retrieval. We adopted a conventional language-based audio retrieval approach, leveraging a joint embedding space for the audio and text encoders trained through contrastive learning. We compared and utilized several state-of-the-art models for the audio encoder, including PaSST, BEATs, VAST, and CAV-MAE. We also employed various datasets with text-audio pairs for training like AudioCaps, WavCaps, Auto-ACD, and MACS. Additionally, we incorporated advanced training techniques such as Mixco and text token masking. During inference, we devised an ensemble method based on queries augmented by ChatGPT. Our final results achieved 39.65 points with a single model and 42.26 points with the ensemble of multiple models in the mean average precision among the top 10 results on the evaluation split of Clotho-V2. Compared with the champion system of the DCASE 2023 Challenge, our model outperformed by 1.09 points for the single mode and 0.84 points for the ensemble of the multiple models, respectively.

***Index Terms*—** Language-based audio retrieval, Audio spectrogram transformer, Data augmentation, Inference time augmentation,

1. INTRODUCTION

The language-based audio retrieval task of the DCASE 2024 Challenge involves building a system that takes a textual query as input and retrieves the corresponding audio from an audio database. The system calculates scores for each audio in the database when a query is given, and ranks them based on the scores. Participants compete on how accurately their system retrieves the target audio. The conventional retrieval model projects the audio and text data into a joint space and uses the similarity of the embeddings as the score for retrieval. The model learns the audio-text relationship with a large number of pairs of audio and corresponding text. A common training method is contrastive learning, where positive pairs of audio-text embeddings have similar values, while negative pairs have less similar values.

To effectively learn the complex text-audio relationship and improve the retrieval score, a large amount of training dataset is required. However, the Clotho-V2 dataset provided by the DCASE challenge contains only about 6000 audio-text pairs and the retrieval performance is limited using only this dataset. Therefore, to achieve high performance in retrieval, it is crucial to leverage pre-trained audio/text encoders trained on large-scale data and use other text-audio pair datasets. The champion system of the DCASE 2023 Challenge leverages both strong pre-trained audio/text encoders and a large amount of external audio-text pair data to improve retrieval performance. The champion system adopted PaSST [1] and RoBERTa [2], the large-scale audio/ text encoders showed superior performance. PaSST is composed of the architecture based on an audio spectrogram transformer [3] and trained with the AudioSet classification task. RoBERTa is an optimized variant of BERT trained by a masked sentence prediction using over 160GB text corpus. Furthermore, contrastive learning with a total of about 450k audio-text pairs, including AudioCaps [4] and WavCaps [5], the total improves the performance significantly.

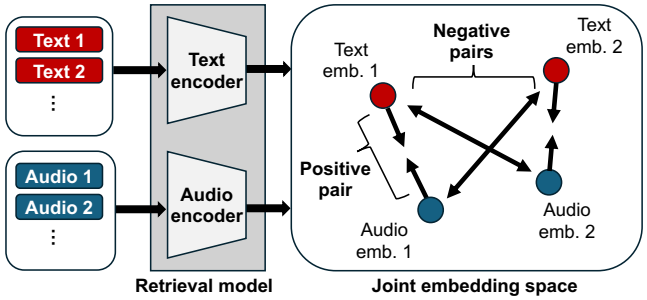

Figure 1: An overview of the conventional language-based audio retrieval system based on contrastive learning. Through contrastive learning, positive pairs of audio-text embeddings have similar values, while negative pairs have less similar values.

As the champion system of the challenge in 2023, we investigated various audio encoders that show state-of-the-art performance in various tasks and datasets containing a large amount of audio-text pairs. For further improvement, we adopt data augmentation including Mix-up contrast [6] and text token masking. In addition, we devised inference time augmentation utilizing a large language model. Third, we devised test time augmentation using the large language model. In this method, we generated paraphrases of the input textual queries and averaged the text embedding of each query. As a result, we achieved 39.65 points with a single model and 42.26 points with the ensemble of multiple models in the mean average precision among the top 10 results (mAP@10), outperforming last year's champion model by 1.08 points and 0.84 points, respectively.

2. LANGUAGE-BASED AUDIO RETRIEVAL MODEL

We describe the training and inference of the retrieval model based on contrastive learning. The retrieval model has audio and text encoders based on neural networks to map the input audio and text onto the joint embedding space. The input audio $\mathbf{X}^{(A)}$ and text

data $\mathbf{X}^{(T)}$ is transformed to D-dimensional embeddings $\mathbf{Z}^{(A)}$ and $\mathbf{Z}^{(T)}$ as follows:

$$\mathbf{Z}^{(A)} = \mathsf{AudioEncoder}(\mathbf{X}^{(A)}), \tag{1}$$

$$\mathbf{Z}^{(T)} = \mathsf{TextEncoder}(\mathbf{X}^{(T)}). \tag{2}$$

The encoders learn the relationship between pairs of audio and text data through contrastive learning. InfoNCE [7] trains the model to discriminate positive and negative from each pair of B audio and text samples. For the embedding of i-th audio and j-th text $\mathbf{Z}_i^{(A)}$ and $\mathbf{Z}_j^{(T)}$, the pairs where $i = j$ are considered positive and the pairs where $i \neq j$ are considered negative. The loss is formulated using the cross-entropy loss with the softmax as follows:

$$\mathcal{L}_{\text{CE}}\left(\mathbf{Z}, \mathbf{Z}'_k, \mathbf{Z}'_:\right) = -\log \frac{\exp\left(\frac{S(\mathbf{Z},\mathbf{Z}'_k)}{\tau}\right)}{\sum_{\mathbf{Z}'\in\mathbf{Z}'_:}\exp\left(\frac{S(\mathbf{Z},\mathbf{Z}')}{\tau}\right)} \tag{3}$$

$$\mathcal{L}^{A\to T}_{\text{infoNCE}} = \sum_i \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A)}, \mathbf{Z}_i^{(T)}, \mathbf{Z}_:^{(T)}\right) \tag{4}$$

$$\mathcal{L}^{T\to A}_{\text{infoNCE}} = \sum_j \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_j^{(T)}, \mathbf{Z}_j^{(A)}, \mathbf{Z}_:^{(A)}\right) \tag{5}$$

$$\mathcal{L}_{\text{infoNCE}} = \mathcal{L}^{A\to T}_{\text{infoNCE}} + \mathcal{L}^{T\to A}_{\text{infoNCE}}, \tag{6}$$

where $\mathbf{Z}'_:$ is a set of $\mathbf{Z}'_1, ..., \mathbf{Z}'_B$, S is the cosine similarity measure, and τ is a trainable temperature parameter.

In the inference, the textual query and audio in the database are projected onto the joint embedding space, and the cosine similarities of the embeddings are measured. The ranking for the retrieval is determined by the order of the similarity from highest to lowest.

3. NETWORK CONFIGURATION OF OUR SYSTEM

3.1. Audio Encoder

We experimented with four types of audio encoder architectures: PaSST [1], BEATs [8], VAST [9], and CAV-MAE [10]. These models are variants of audio spectrogram transformers (ASTs) [3] that apply Vision Transformers [11] to audio spectra.

PaSST enhances the AST by incorporating regularization and speed improvements through patch-out techniques. Additionally, it employs distinct positional encodings for the time and frequency dimensions, leading to performance enhancements. For our experiments, we utilized the weights pre-trained on the AudioSet classification task[1]. The stride size for the frequency and time was 16, i.e., the patches were not overlapped.

CAV-MAE extends the AST into an audio-visual model by integrating the outputs of AST and a Video Transformer. This combined output is fed into a subsequent transformer that captures the interrelationships between audio and visual modalities through self-attention mechanisms. The weights used for CAV-MAE were pre-trained using a multi-task loss that combines contrastive learning and masked autoencoder loss on both AudioSet and VGGSound datasets. For our experiments, we utilized the scale++ model pre-trained on self-supervised learning using AudioSet [2].

BEATs introduces a novel discrete audio tokenizer to the AST framework, leveraging self-supervised learning to achieve high performance. This model iteratively trains the AST-based SSL model and the acoustic tokenizer, resulting in significant performance improvements. Notably, BEATs demonstrated its high efficacy by being employed in the winning methodology for the DCASE 2023 Task 6 captioning task. For our experiments, we utilized the weights pre-trained on the AudioSet classification task[3].

VAST is a multi-modal model that integrates vision, audio, subtitles from videos, and texts into a unified framework. It is trained on the VAST-27M dataset, which includes 27 million video clips with captions generated for each modality. The model leverages these captions to support various tasks such as retrieval, captioning, and question-answering. VAST has demonstrated state-of-the-art performance on multiple cross-modality benchmarks. For our experiments, we utilized the weights pre-trained on only the self-supervised learning and weight that was fine-tuned with the audio captioning task, respectively [4].

3.2. Text Encoder

We utilized RoBERTa [2] as our text encoder. RoBERTa, an optimized variant of BERT, focuses solely on the Masked Language Model (MLM) objective and removes the Next Sentence Prediction (NSP) objective. It is pre-trained on a large, diverse corpus of 160GB. RoBERTa employs longer training times and larger batch sizes for enhanced performance. We used publicly available pre-trained weights to capture semantic information effectively. Notably, RoBERTa was used in the winning method for the 2023 audio retrieval task. In preliminary experiments, we also tested BERT, Sentence-BERT, and T5, but RoBERTa outperformed them all, leading us to select RoBERTa exclusively.

4. DATASET AND AUGMENTATION

4.1. Datasets

We utilized several datasets for our experiments: Clotho, Clotho-GPT [12, 13], MACS [14], AudioCaps [4], WavCaps [5], and Auto-ACD [15].

Clotho-V2 contains audio recordings ranging from 10 to 30 seconds in length. The development set is divided into training, validation, and test splits with 3,840, 1,045, and 1,045 recordings, respectively. Each audio recording in the dataset is associated with five human-written captions, each between 8 and 20 words long.

Clotho-V2-GPT is an augmented version of Clotho v2, where the original human-written captions are expanded using OpenAI's GPT. This dataset includes 96,000 captions generated by GPT based on the original audio's captions and keywords from metadata. Since this dataset is only for the training split, we generated Clotho-GPT for the validation and evaluation split by following the instructions provided by the authors [13].

MACS is extracted from the TAU Urban Acoustic Scenes 2019 and contains approximately 3,900 samples, each 10 seconds long, totaling around 47 hours of audio. Captions are manually created, with roughly five captions per audio clip. The vocabulary size is 2,803 words.

AudioCaps is derived from AudioSet and contains approximately 53,000 samples, totaling around 150 hours of audio. The majority of the clips are 10 seconds long. The captions are manually created, with one caption per audio clip. The vocabulary size is 5,129 words.

[1] https://github.com/kkoutini/passt_hear21

[2] https://github.com/YuanGongND/cav-mae

[3] https://github.com/microsoft/unilm/tree/master/beats

[4] https://github.com/TXH-mercury/VAST

WavCaps includes samples from FreeSound, BBC Sound Effects, SoundBible, and AudioSetSL, totaling around 400,000 samples. FreeSound contributes over 260,000 samples, while AudioSetSL provides over 100,000 samples. The clip lengths vary from 10 seconds to several minutes, with an average length of 67 seconds, totaling approximately 7,500 hours of audio. Not all samples are used during training. Captions are automatically generated using GPT based on existing metadata (tags, etc.). The prompts are tailored for each source dataset. Each audio clip has one caption, with a vocabulary size of 28,721 words.

Auto-ACD comprises samples from AudioSet and VGGSound. According to the paper, the subset from VGGSound performs better on Clotho, so it is primarily used. It contains 1.9 million samples, with 180,000 samples from VGGSound. Most clips are 10 seconds long, totaling approximately 500 hours of audio. Captions are generated using ChatGPT, leveraging existing tags and object recognition results from videos. Each audio clip has one caption, with a vocabulary size of 8,157 words. Since a subset using VGGSound shows better performance for the Clotho dataset as reported in [15], we only used this subset.

4.2. Data Augmentation

Mix-up contrast (Mixco) [6] is a data augmentation method originally applied to text-to-image contrastive learning methods. This method trains the model using the semi-positive pairs consisting of the mixed input images and the corresponding texts. By relaxing the discrimination problem of contrastive learning using semi-positive pairs, the model learns better representation. To apply this method to language-based audio retrieval, we mix the i-th and audio in the batch $\mathbf{X}_i^{(A)}$ and another audio $\mathbf{X}_{\phi(i)}^{(A)}$ in the waveform and transform it as follows:

$$\mathbf{X}_i^{(A')} = \lambda \mathbf{X}_i^{(A)} + (1-\lambda)\mathbf{X}_{\phi(i)}^{(A)}, \tag{7}$$

$$\mathbf{Z}_i^{(A')} = \mathsf{AudioEncoder}(\mathbf{X}_i^{(A')}), \tag{8}$$

where $\phi(i)$ is a randomly selected index for i and $\lambda \in (0,1)$ is a random variable sampled from the uniform distribution. From the embeddings of the mixtures, the additional loss of Mixco is obtained by the weighted sum of the infoNCE loss to discriminate semi-positive and negative pairs similar to Eq. (4) and Eq. (5) as follows:

$$\begin{aligned}\mathcal{L}_{\text{mixco}}^{A\to T} = \sum_i \lambda \Big\{ &\mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A')}, \mathbf{Z}_i^{(T)}, \mathbf{Z}_:^{(T)}\right) \\ &+(1-\lambda)\ \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_i^{(A')}, \mathbf{Z}_{\phi(i)}^{(T)}, \mathbf{Z}_:^{(T)}\right)\Big\}, \end{aligned} \tag{9}$$

$$\begin{aligned}\mathcal{L}_{\text{mixco}}^{T\to A} = \sum_j \Big\{ &\lambda\mathcal{L}_{\text{CE}}\left(\mathbf{Z}_j^{(T)}, \mathbf{Z}_j^{(A')}, \mathbf{Z}_:^{(A')}\right) \\ &+(1-\lambda)\ \mathcal{L}_{\text{CE}}\left(\mathbf{Z}_{\phi(j)}^{(T)}, \mathbf{Z}_j^{(A')}, \mathbf{Z}_:^{(A')}\right)\Big\}, \end{aligned} \tag{10}$$

$$\mathcal{L}_{\text{mixco}} = \mathcal{L}_{\text{mixco}}^{A\to T} + \mathcal{L}_{\text{mixco}}^{T\to A}. \tag{11}$$

We used the same temperature parameter for Eq. (3). In our experiment, we used the combination of the original info NCE loss and Mixco loss:

$$\mathcal{L} = \mathcal{L}_{\text{infoNCE}} + \mathcal{L}_{\text{mixco}}. \tag{12}$$

Text token masking is a data augmentation method for the input text. The text tokens are randomly replaced with `[MASK]` token. We set the replace probability to 10%. We experimentally confirmed that this method improved the infoNCE loss of the validation set.

4.3. Inference Time Augmentation

We devised an inference time query augmentation method using a large language model. This method improves the retrieval performance by generating additional queries using a large language model, thereby supplementing the linguistic information of the original query. The text encoder projects the original and additional queries and then the embeddings of each query are averaged. To generate additional queries, we used OpenAI's GPT and the same prompt used in Clotho-V2-GPT.

5. EXPERIMENT

5.1. Experimental Setting

We trained seven models in different conditions, including the audio encoder, batch size, and the optimizing procedure (see Table 1.) All models were trained with Clotho, AudioCaps, WavCaps, and Auto-ACD. In addition, the models "A" and "G" were trained with MACS, and the model "G" was trained with Clotho-V2-GPT in place of the original Clotho-V2. The replacement probability of the original caption to the generated caption of Clotho-V2-GPT was set to 0.3 the same as [13]. The preprocess and sampling rate of the audio followed the original implementation of each audio encoder. We set the number of the dimension of the joint embedding space to 1024. We optimized the model using Adam [16] and AdamW [17]. The learning rate was changed by iterations using a cosine scheduler with 1 or 2 warm-up epoch and the maximum learning rate was 1×10^{-5}. The initial value of the temperature parameter τ used in Eq (3) was 0.02. To avoid learning unexpected relationships between the audio and text caused by the difference among the datasets, we generated each batch from the same dataset. In the training, we conducted validation by 20% of each epoch and saved the model weight. After training, the weights of the models that achieved the top 10 in validation mAP@10 were averaged to form the final model weights. For the inference time augmentation, we generated five additional captions for Clotho-V2 and the evaluation dataset of this challenge. For the ensemble of the models, we obtained the embeddings from the model with and without inference time augment, i.e., two embeddings were obtained from the single model.

5.2. Result

We evaluated our models using the test split of the Clotho-V2 dataset. Table 1 shows the mAP@10 of the single models. Model "A" was the best in our models achieving 39.65 points outperforming the champion system of the DCASE 2023 Challenge by 1.09 points. In addition, the performance of the models "B" and "C" with Mixco or text token masking was comparable to the model "A". In terms of the audio encoder, the models "F" and "G" with VAST achieved over 39 points. In contrast, the performance of the model "D" with CAV-MAE did not exceed 38 points. Table 2 shows the mAP@10 of the ensemble of the models. The ensembles significantly improved the retrieval performance compared to the single models and both ensembles outperformed the champion system of the challenge in 2023. In particular, the second was better and outperformed the champion system by 0.84 points. A possible reason

Table 1: Performance of the single models. "captioning" and "vanilla" for VAST refers to the pre-training including fine-tuning for the audio captioning task or not.

Model ID	Audio encoder	Data augmentation		Dataset		Optimizing procedure				mAP@10
		Mixco	Token masking	MACS	Clotho-V2 -GPT	optimizer	batch size	epochs	warm-up epochs	
A	PaSST	-	-	✓	-	Adam	256	15	2	39.65
B	PaSST	✓	-	-	-	Adam	128	15	2	39.28
C	PaSST	-	✓	-	-	Adam	256	15	2	39.09
D	CAV-MAE	-	-	-	-	Adam	256	15	2	37.95
E	BEATs	-	-	-	-	Adam	64	15	1	38.62
F	VAST (captioning)	-	-	-	-	Adam	64	15	1	39.04
G	VAST (vanilla)	✓	✓	✓	✓	AdamW	256	10	2	39.10
The champion system of DCASE 2023 Challenge										38.56
The baseline system										22.20

Table 2: Performance of the ensembles of the multiple models

Submission ID	Used model ID	mAP@10
1	A, B, C, D, E, F	42.20
2	A, B, C, F, G	42.26
The champion system of DCASE 2023 Challenge		41.42

for the significant improvement by the ensemble is that the different audio encoders capture the different characteristics of the audio and compensate for each result. The first ensemble included the models with four different audio encoders and the second included the models with PaSST or VAST. This result implies that choosing a model with a highly effective encoder is important, and simply combining models with various encoders is not sufficient.

6. CONCLUSION

This report shows our language-based audio retrieval system submitted to the DCASE 2024 Challenge. We trained multiple models with various audio encoders and datasets containing a large amount of audio-text pairs. The best performance of the single model and the ensemble model outperformed the champion model of the DCASE 2023 Challenge by 1.09 and 0.84 points, respectively.

7. REFERENCES

[1] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," in *Proc. INTERSPEECH*, 2022.

[2] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized bert pretraining approach," 2019.

[3] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," in *Proc. INTERSPEECH*, 2021.

[4] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. NAACL*, 2019.

[5] X. Mei, C. Meng, H. Liu, Q. Kong, T. Ko, C. Zhao, M. D. Plumbley, Y. Zou, and W. Wang, "WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[6] S. Kim, G. Lee, S. Bae, and S.-Y. Yun, "Mixco: Mix-up contrastive learning for visual representation," *arXiv preprint arXiv:2010.06300*, 2020.

[7] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018. [Online]. Available: https://arxiv.org/abs/1807.03748

[8] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. ICML*, 2023.

[9] S. Chen, H. Li, Q. Wang, Z. Zhao, M. Sun, X. Zhu, and J. Liu, "VAST: A vision-audio-subtitle-text omni-modality foundation model and dataset," in *Proc. NeurIPS*, 2024.

[10] Y. Gong, A. Rouditchenko, A. H. Liu, D. Harwath, L. Karlinsky, H. Kuehne, and J. R. Glass, "Contrastive audio-visual masked autoencoder," in *Proc. ICLR*, 2022.

[11] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. ICML*, 2021, pp. 10 347–10 357.

[12] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. ICASSP*, 2020, pp. 736–740.

[13] P. Primus, K. Koutini, and G. Widmer, "CP-JKU's submission to task 6b of the dcase2023 challenge: Audio retrieval with PaSST and GPT-augmented captions," DCASE2023 Challenge, Tech. Rep., June 2023.

[14] I. Martín-Morató, A. Mesaros, T. Heittola, T. Virtanen, M. Cobos, and F. J. Ferri, "Sound event envelope estimation in polyphonic mixtures," in *Proc. ICASSP*, 2019.

[15] L. Sun, X. Xu, M. Wu, and W. Xie, "A large-scale dataset for audio-language representation learning," *arXiv preprint arXiv:2309.11500*, 2023.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[17] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.