ALIGNING AUDIO AND TEXT EMBEDDINGS FOR THE LANGUAGE-BASED AUDIO RETRIEVAL TASK OF THE DCASE CHALLENGE 2022

Technical Report

Benno Weck[1,2]*, Miguel Pérez Fernández*[1,2]*, Holger Kirchhoff*[1]*, Xavier Serra*[2]

[1] Huawei Technologies, Munich Research Center, Germany
{firstname.lastname}@huawei.com
[2] Universitat Pompeu Fabra, Music Technology Group, Spain
{firstname.lastname}01@estudiant.upf.edu,xavier.serra@upf.edu

ABSTRACT

Our challenge submission shows how large-scale pretrained deep learning models can serve as a strong basis for a cross-modal (text-to-audio) retrieval system. Our system uses embeddings extracted by these models in a general alignment framework to connect matching pairs of audio and text. It processes audio and text separately through different pretrained models, each returning an embedding. Shallow neural networks map the embeddings to a common dimensionality. The cross-modal alignment of the individual embeddings is optimised using a contrastive loss. We employ the RoBERTa foundation model as the text embedding extractor. A pretrained PANNs model extracts the audio embeddings. The embedding extractor model weights remain frozen. To improve the generalisation of our model, we investigate how pretraining with audio and associated noisy text collected from the online platform Freesound improves the performance of our method. We find that a two-stage training process consisting of pretraining with noisy data and fine-tuning with the challenge datasets gives the best results for our approach. Our system showcases a simple yet effective method which is superior to the challenge baseline.

1. INTRODUCTION

The DCASE2022 challenge subtask 6b provides a platform to stimulate research in the underexplored problem domain of language-based audio retrieval [1]. The goal of this task is to find the closest matching audio recordings for a given text query. A possible application for this task is a search engine for audio files in which a user can enter a free-form textual description to retrieve matching recordings. Such systems need to draw a connection between the two modalities: audio and text.

Given the complex nature of both audio and text, we expect that a submission for this task can only be competitive if it can capitalise on a large amount of training data. Due to the novelty of the task, not many previous studies and systems exist for language-based audio retrieval and training data is still limited. We instead turn to the fields of machine listening, specifically audio tagging, and natural language processing to draw inspiration from related problems and make use of existing resources such as pretrained models. It has become a popular approach to use large-scale pretrained models in a transfer learning setup for tasks where only limited training data is available.

The goal of this work is to build a simple, generic cross-modal alignment system that leverages the power of pretrained models to semantically connect audio and text. With the help of a metric learning framework we intend to link the two modalities. Our system should be able to process audio and text independently to be used in a cross-modal retrieval context. We aim to limit the complexity of our approach by employing the pretrained models with fixed weights and only training shallow network architectures to perform the alignment.

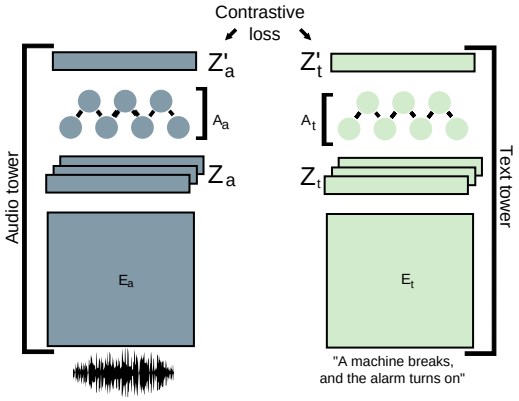

Figure 1: Overview of the architecture of our system. An audio tower and a text tower process the respective input data separately and produce a single embedding.

2. METHOD

In our approach, we adopt a metric learning framework to embed the audio and text into a shared acoustic-textual space. Our system consists of two components – an audio tower and a text tower – to separately process the audio and text input. Each tower is further divided into an encoder, $\mathrm{E}(\cdot)$, and an embeddings' adapter, $\mathrm{A}(\cdot)$. As the audio encoder E_a and the text encoder E_t, we employ pretrained models. We do not fine-tune the encoder models in our approach and only optimise the adapters. An overview of our method is presented in Figure 1.

More specifically, an audio input $\mathbf{X}_a$ or a text input $\mathbf{X}_t$ are

processed by E_a and E_t, respectively, as

$$\begin{aligned} \mathbf{Z}_a &= \mathrm{E}_a(\mathbf{X}_a), \\ \mathbf{Z}_t &= \mathrm{E}_t(\mathbf{X}_t), \end{aligned} \tag{1}$$

where $\mathbf{Z}_i \in \mathbb{R}^{T_i \times F_i}, i \in \{a, t\}$ is a sequence of T_i intermediate representations with F_i features provided by the pretrained model (i.e., an embedding sequence). Then, the adapters A_a and A_t will process $\mathbf{Z}_a$ and $\mathbf{Z}_t$ as

$$\begin{aligned} \mathbf{Z}'_a &= \mathrm{A}_a(\mathbf{Z}_a), \\ \mathbf{Z}'_t &= \mathrm{A}_t(\mathbf{Z}_t), \end{aligned} \tag{2}$$

where $\mathbf{Z}'_a, \mathbf{Z}'_t \in \mathbb{R}^{F'}$ are single embeddings and F' denotes their dimensionality. The intermediate embedding sequences $\mathbf{Z}_a$ and $\mathbf{Z}_t$ produced by the audio and text encoder respectively will differ in dimensionality. The main purpose of the adapters is to match the dimensionality of text and audio embeddings in order to enable comparisons.

In our work, we use a contrastive loss [2] to align the embedded spaces. With the contrastive loss, all the samples belonging to the same reference label l (e.g., a matching text and audio pair) are pulled together in the embedded space, while being pushed away from samples belonging to other labels (e.g., two different audio files or a non-matching text and audio pair).

Given the cosine similarity s between a pair of embeddings with labels l_1 and l_2, the contrastive loss is defined by:

$$L_{contrastive} = \begin{cases} 1 - s & \text{if } l_1 = l_2 \\ \max(0, s) & \text{otherwise.} \end{cases} \tag{3}$$

We compute the loss for every possible combination of similar and dissimilar samples (including text-to-text and audio-to-audio pairs) and take the mean across all non-zero loss values.

For the final application as a text-to-audio retrieval system, we compute the embedding of the text query $\mathbf{Z'}_t$ and compares it to all pre-computed embeddings $\mathbf{Z'}_a$ of the audio items in the dataset by means of the cosine similarity. Ranking the audio items by their similarity score in descending order provides the retrieval results.

3. EXPERIMENTS

3.1. Datasets

As the main dataset in our work, we employ the development dataset provided for this challenge, *Clotho v2* [3], and use its official splits for training, validation, and final evaluation (testing). We posit that the Clotho dataset is relatively small for the training of deep-learning-based retrieval systems and any system might benefit from additional training data. Datasets combining audio and text are scarce, however, and the few that exist next to Clotho are either specific to a certain domain (e.g., urban soundscapes only [4]) or their audio content is not freely accessible [5]. This is why we decided to use weakly aligned text and audio pairs collected from the online platform Freesound [6], which also served as the data source for Clotho. Freesound allows users to upload an audio recording along with a textual description and a set of tags. This type of metadata was used before to extend the training data of Clotho but in the context of an automated audio captioning task [7]. For simplicity and reproducibility, we limit ourselves to the *dev* subset of the *FSD50k* dataset [8]. We assume that the audios in this dataset closely resemble the challenge audio data as the dataset mainly comprises recordings of sound events. Moreover, similarly to Clotho, audio clips are not longer than 30 seconds. The descriptions and tags in the dataset contain rich information about the content of the audio clip as can be seen from the examples given in Table 1. Nevertheless, the text data is noisy and also contains some undesired text.[1] To clean the descriptions we remove all HTML mark-up and limit each text to 500 characters in a pre-processing step. To form a "sentence" out of the tags, we join them with a single white space in the order given by the content uploader. The *dev* split of the FSDK50 dataset contains almost 44100 files and we use half of them. By using descriptions and tag sequences, we can extend the training data by 40966 text-audio pairs (more than twice the amount of caption-audio pairs in the training subset of Clotho). We refer to the data from Clotho as "clean" and from FSD50k as "noisy".

Description	Tags
"Typing on a mechanical keyboard"	"click", "keyboard", "mechanical", "computer", "typing", "button"
"Pouring liquid in a shot glass, picking it up, drinking & slamming it down (not too hard) on the table."	"slam", "glass", "pour", "drink", "liquid", "alcohol", "shot"
"opening of shower curtain, turning shower on, water running, turning shower off, getting out"	"shower", "water", "bathroom", "bathtub", "human"

Table 1: Hand-picked examples of descriptions and text labels from the metadata of the FSD50k dataset.

3.2. Evaluation & Metrics

We evaluate the ranked retrieval results generated by our systems with the same four metrics as the challenge organisers. Specifically, we report three 'recall at k' metrics (*Recall@1*, *Recall@5*, *Recall@10*) and one 'mean average precision at k' (*mAP@10*), where a score for a given query is computed for the top-k retrieved results and all scores are averaged over the entire set of queries. We direct the reader to [9] for an in-depth explanation of the metrics.

3.3. Implementation details

Our system is implemented by relying on the *PyTorch* [10] framework in connection with the *pytorch-metric-learning* package [11]. For the text processing, we employ the *Transformers* library [12] and use the pretrained *distilroberta-base* model as the text encoder. This model is a compressed version of the original *RoBERTa* model [13] created by a knowledge distillation procedure [14]. It is smaller and faster than the original variant while retaining high performance on downstream tasks. Similar to our previous work on audio captioning [15], we decided to use the penultimate layer as the intermediate embeddings $\mathbf{Z}_t$. The extracted text embeddings have a dimensionality F_t of 768.

For the audio processing, we use a pretrained *PANNs* model [16] as the audio encoder. We follow the authors' suggestion and

[1] For example: "CAUTION: THIS PACK IS A CHEAP HOME RECORD. (But this one sounds a bit better)"

	Recall@1	Recall@5	Recall@10	mAP@10
Challenge baseline	0.03 (0.03 - 0.04)	0.11 (0.10 - 0.12)	0.19 (0.18 - 0.20)	0.07 (0.06 - 0.07)
ATAE	0.071 (0.064 - 0.078)	0.217 (0.206 - 0.228)	**0.325** (0.312 - 0.337)	0.136 (0.128 - 0.143)
ATAE-ET	0.064 (0.057 - 0.070)	0.194 (0.184 - 0.205)	0.288 (0.275 - 0.300)	0.121 (0.114 - 0.128)
ATAE-EP-F	0.067 (0.061 - 0.074)	0.200 (0.189 - 0.210)	0.299 (0.286 - 0.311)	0.127 (0.120 - 0.134)
ATAE-NP-F	**0.072** (0.065 - 0.079)	**0.225** (0.214 - 0.236)	**0.325** (0.313 - 0.338)	**0.139** (0.131 - 0.146)

Table 2: Retrieval metrics for the four submitted systems and the challenge baseline on the development evaluation dataset. The 95% confidence intervals computed by jackknife resampling are given in parentheses. The highest value for each metric is marked in bold.

compute embeddings by taking the post-activation output of the penultimate layer of their *CNN14* model.[2] All audio clips are resampled to a sampling rate of 32 kHz in a preprocessing step. The extracted intermediate audio embeddings $\mathbf{Z}_a$ have a dimensionality F_a of 2048.

We use simple feed-forward neural networks to adapt each embedding sequence to the common dimensionality. Both adapters consist of a two-layer perceptron with a layer size of 512 and a rectified linear unit (ReLU) as activation function after the first layer. We use the average of all embeddings in a sequence as the final representation.

The system is optimised by minimising the contrastive loss with the Adam algorithm [17] ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$). To form a minibatch we randomly select 32 audio-text pairs from the training set. Every epoch the mAP@10 metric is computed on the validation dataset. The training is stopped if no improvement was found for 10 epochs and the model weights are reverted to the checkpoint of the epoch with the highest score.

3.4. Submitted systems

We submit four different configurations of our system. All share the same model hyperparameter configurations but differ in the way the available training data was used to train them. Specifically, we experiment with:

1. adding no external dataset in our training,
2. extending the training data with noisy data from the FSD50k dataset,
3. pretraining with noisy and clean data and later fine-tuning with clean data only,
4. and pretraining exclusively with noisy data and fine-tuning with clean data only.

In every training (also if we refer to it as pretraining or fine-tuning), we follow the optimisation procedure described above.

ATAE: Aligned Text and Audio Embeddings In its standard configuration, our system is trained solely with the challenge development dataset Clotho. We refer to it as "Aligned Text and Audio Embeddings" or *ATAE* for short.

ATAE-ET: Aligned Text and Audio Embeddings – Extended dataset for Training Next, we want to investigate if adding extra training data helps to improve retrieval performance. To achieve this we combine the noisy FSD50k and the clean Clotho data into a single training dataset.

[2]Pretrained weights can be found at: `https://doi.org/10.5281/zenodo.3987831`

ATAE-EP-F: Aligned Text and Audio Embeddings – Extended dataset for Pretraining – Fine-tuning To balance out the potential negative effects of the noise in the training data, we fine-tune the trained ATAE-ET model by again training with the clean Clotho dataset.

ATAE-NP-F: Aligned Text and Audio Embeddings – Noisy dataset for Pretraining – Fine-tuning Finally, to be able to better judge the effect of the noisy data for pretraining, we use the datasets in two separate training stages. We first train a model on the noisy data and then fine-tune it on the clean dataset.

4. RESULTS

Table 2 compares the metrics achieved on the challenge development test set for our four systems with the challenge baseline. We follow the lead of the challenge organisers and report a jackknife approximated 95% confidence interval for each metric [18]. Based on the results, we make the following observations. First, our approach produces good quality results even in the standard training setup (ATAE). Second, extending the challenge dataset with additional (noisy) training data (ATAE-ET) significantly degrades retrieval performance. Third, even fine-tuning the second system on the clean challenge dataset seems to give worse results in comparison with simply training only with the challenge dataset (ATAE). Fourth, our system first pretrained with noisy data only and then fine-tuned on the challenge dataset (ATAE-NP-F) improves on the performance of the first experiment but only slightly. Finally, all of our submitted systems surpass the challenge baseline in each metric by a comfortable margin.

Since the metrics of our best system lie within the confidence intervals of the next best system and vice versa, we conclude that no significant difference is measurable between them. These results lead us to the conclusion that no apparent advantage exists for our method in utilising additional noisy training data, as was hoped.

5. CONCLUSION

We presented our submission for the *Language-based Audio Retrieval* subtask of the DCASE2022 challenge. Our approach consists of extracting embeddings for the text and the audio through pretrained encoder models and mapping these embeddings to a shared space with a cross-modal alignment procedure. We achieve the best results on the development test set with a model pretrained with noisy text-audio data collected from a Freesound dataset. However, we did not find a significant improvement in comparison to a model that was trained only using the challenge development dataset.

6. REFERENCES

[1] H. Xie, S. Lipping, and T. Virtanen, "DCASE 2022 Challenge Task 6B: Language-Based Audio Retrieval," 2022. [Online]. Available: https://arxiv.org/abs/2206.06108

[2] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised Contrastive Learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 18 661–18 673. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/d89a66c7c80a29b1bdbab0f2a1a94af8-Paper.pdf

[3] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an Audio Captioning Dataset," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Barcelona, Spain: IEEE, May 2020, pp. 736–740. [Online]. Available: https://ieeexplore.ieee.org/document/9052990/

[4] I. Martín-Morató and A. Mesaros, "Diversity and Bias in Audio Captioning Datasets," in *Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events 2021 (DCASE 2021), Online, November 15-19, 2021*, F. Font, A. Mesaros, D. P. W. Ellis, E. Fonseca, M. Fuentes, and B. Elizalde, Eds., 2021, pp. 90–94. [Online]. Available: https://dcase.community/documents/workshop2021/proceedings/DCASE2021Workshop_Martin_34.pdf

[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating Captions for Audios in The Wild," in *NAACL-HLT*, 2019.

[6] F. Font, G. Roma, and X. Serra, "Freesound Technical Demo," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 411–412, event-place: Barcelona, Spain. [Online]. Available: https://doi.org/10.1145/2502081.2502245

[7] Q. Han, W. Yuan, D. Liu, X. Li, and Z. Yang, "Automated Audio Captioning with Weakly Supervised Pre-Training and Word Selection Methods," in *Proceedings of the 6th Workshop on Detection and Classification of Acoustic Scenes and Events 2021 (DCASE 2021), Online, November 15-19, 2021*, F. Font, A. Mesaros, D. P. W. Ellis, E. Fonseca, M. Fuentes, and B. Elizalde, Eds., 2021, pp. 6–10. [Online]. Available: https://dcase.community/documents/workshop2021/proceedings/DCASE2021Workshop_Han_9.pdf

[8] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "FSD50K: an open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022, publisher: IEEE.

[9] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. New York: Cambridge University Press, 2008.

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d. Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[11] K. Musgrave, S. Belongie, and S.-N. Lim, "PyTorch Metric Learning," 2020. [Online]. Available: https://arxiv.org/abs/2008.09164

[12] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[13] 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. [Online]. Available: https://arxiv.org/abs/1907.11692

[14] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[15] B. Weck, X. Favory, K. Drossos, and X. Serra, "Evaluating Off-the-Shelf Machine Listening and Natural Language Models for Automated Audio Captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, Nov. 2021, pp. 60–64.

[16] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[18] A. Mesaros, A. Diment, B. Elizalde, T. Heittola, E. Vincent, B. Raj, and T. Virtanen, "Sound Event Detection in the DCASE 2017 Challenge," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 992–1006, 2019.