LANGUAGE-BASED AUDIO RETRIEVAL TASK IN DCASE 2022 CHALLENGE

Huang Xie, Samuel Lipping, Tuomas Virtanen

Audio Research Group, Tampere University, Tampere, Finland

ABSTRACT

Language-based audio retrieval is a task, where natural language textual captions are used as queries to retrieve audio signals from a dataset. It has been first introduced into DCASE 2022 Challenge as Subtask 6B of task 6, which aims at developing computational systems to model relationships between audio signals and free-form textual descriptions. Compared with audio captioning (Subtask 6A), which is about generating audio captions for audio signals, language-based audio retrieval (Subtask 6B) focuses on ranking audio signals according to their relevance to natural language textual captions. In DCASE 2022 Challenge, the provided baseline system for Subtask 6B was significantly outperformed, with top performance being 0.276 in mAP@10. This paper presents the outcome of Subtask 6B in terms of submitted systems' performance and analysis.

Index Terms— Language-based audio retrieval, DCASE 2022 Challenge, Clotho.

1. INTRODUCTION

With the growth of multimedia content in recent decades, there is a need for retrieval methods that can efficiently organize the data based on its content, and retrieve relevant items when doing searches to datasets. Natural language provides an efficient way to represent complex information about multimedia. It can represent high level information about data that goes beyond any fixed taxonomies. For audio signals, natural language can represent information related to temporal relationships between sound sources, and attributes of sounds and their environment.

Language-based multimedia retrieval has received increasing attention in recent years. The majority of recent works has focused heavily on the visual domain [1, 2]. For example, there are plenty of approaches [3] tackling content-based image retrieval with free-form textual descriptions. In contrast, only a few studies have been conducted on language-based audio retrieval in the existing literature. Early works [4, 5] deal with language-based audio retrieval using multi-word text queries consisting of audio tags or class labels, rather than sentence-like textual descriptions, e.g., captions. Recent studies [6, 7] prompt research in this field by exploring human written captions as queries. In DCASE 2022 Challenge, language-based audio retrieval is introduced into as Subtask 6B, which aims to inspire further research into audio retrieval with unconstrained textual descriptions.

In this paper, we present the task setup and submissions for Subtask 6B of task 6 in DCASE 2022 Challenge. We introduce the datasets for system development and evaluation, describe the baseline system for Subtask 6B, and present the challenge submissions. Evaluation and analysis of submitted systems includes general statistics on systems and performance and system characteristics.

The remainder of this paper is organized as follows. In Section 2, we describe the task setup, including task description, task datasets, and evaluation metrics. Then, we introduce the task baseline system in Section 3. We present the evaluation results and analysis of challenge submissions in Section 4. Finally, we conclude this paper in Section 5.

2. TASK SETUP

In this section, we introduce the task description, datasets for system development and evaluation, and evaluation metrics.

2.1. Task Description

Language-based audio retrieval is concerned with retrieving audio signals using their sound content textual descriptions (i.e., audio captions). With this task, the goal is to evaluate audio retrieval methods, where a retrieval system takes an audio caption as a text query and ranks audio signals in a fixed dataset according to their relevance to the caption. In DCASE 2022 Challenge, human-written natural language audio captions are used as text queries. For each query, the retrieval task is to retrieve 10 audio files from a given evaluation dataset and sort them according to their relevance to the query.

2.2. Development Dataset

The Clotho v2 [8] is provided as the task development dataset, which consists of audio samples of 15 to 30 seconds duration, with each audio sample having five captions of eight to 20 words length. There are 6,974 audio samples with 34,870 captions in total. All audio samples are sourced from the Freesound platform [9], and captions are crowd-sourced using a three-step framework [8].

The Clotho v2 [8] is divided into a training split of 3,839 audio clips with 19,195 captions, a validation split of 1,045 audio clips with 5,225 captions, and a testing split of 1,045 audio clips with 5,225 captions. These splits are created by first constructing the sets of unique words of the captions of each audio clip. These sets of words are combined to form the bag of words of the whole dataset, from which the frequency of a given word can be derived. With the unique words of audio files as classes, multi-label stratification is applied. The data collecting procedure is explained in detail in [8].

2.3. Clotho Retrieval Evaluation Dataset

The task evaluation dataset consists of 1,000 audio samples sourced from the Freesound platform [9], and one human written caption is provided for each audio sample. The audio samples are collected following the procedure described in [8], by optimizing the tag distribution of the selected samples. The samples were selected from the set of files not used in Clotho v2. Captions were gathered using

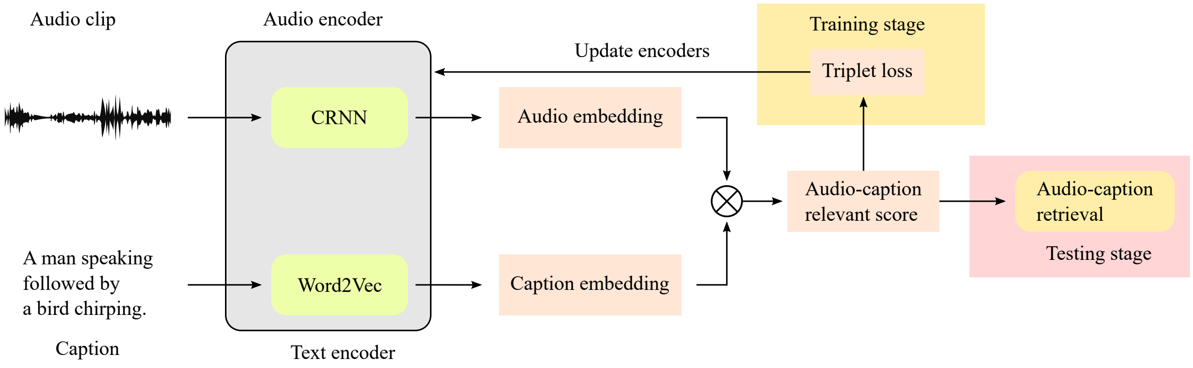

Figure 1: The baseline system for the language-based audio retrieval subtask.

Dataset	Split	#Audio	#Captions
Development	Training	3839	19195
	Validation	1045	5225
	Testing	1045	5225
Evaluation		1000	1000

Table 1: Statistics of the development and evaluation datasets.

the first crowd-sourcing step of Clotho v2 manually screened for typographical errors and speech transcription. Table 1 summarizes the information about the development and evaluation datasets.

2.4. Evaluation Metrics

In the evaluation, the ground truth relevance of audio samples are considered binary (i.e., only the audio samples belong to the caption query are considered relevant, and all the others not relevant). The submissions for this task will be evaluated using mean average precision at top-10 (mAP@10) as the main metric, and recall at k (R@k with $k \in \{1, 5, 10\}$) as the secondary metrics.

The mAP metric has been widely used for evaluating the performance of cross-modal retrieval algorithms [3]. It is a rank-aware metric, which 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. The more relevant items in the top rank list, the higher mAP value it has. The R@k metric is another standard, rank-unaware retrieval metric [6], which is defined as the proportion of relevant items among the top-K retrieved results to all the relevant items in the evaluation dataset, averaged across all the caption queries. The challenge submissions will be ranked by the mAP@10 metric.

3. BASELINE SYSTEM

In this section, we describe the task baseline system, as illustrated in Figure 1. The baseline system is a simplified version of the audio-text aligning framework presented in [7], which calculates relevant scores between encoded textual descriptions (i.e., encoded captions) and encoded audio signals. It consists of two input encoders: one for audio, and the other for text, as illustrated in Figure 1. These two modality-specific encoders generate vector representations (i.e., audio embeddings and caption embeddings) for audio clips and textual descriptions. Then, the relevance score between an audio clip and a textual description is calculated by the dot product of their vector representations. The baseline system is optimized with a sampling-based triplet loss [19] at the training stage, and then applied to retrieve audio for caption queries at the testing stage.

3.1. Audio Encoder

A convolutional recurrent neural network (CRNN) [20] is used as the audio encoder, which extracts frame-wise acoustic embeddings from audio signals. It consists of five convolution blocks, followed by a bidirectional gated recurrent unit (BiGRU). Each convolution block includes an initial batch normalization, a convolutional layer with padded 3×3 convolutions, and a LeakyReLU activation with a slope of -0.1. After the first, third, and fifth convolution blocks, one L4-Norm subsampling layer is used to reduce the temporal dimension of each block's output by a factor of four. A dropout layer with a rate of 0.3 is placed between the last L4-Norm layer and the BiGRU. Lastly, an up-sampling operation is applied to ensure the final output has the same temporal dimension as the CRNN input.

The CRNN audio encoder takes 64-dimensional log mel-band energies as input. Each audio clip is split into 40 ms Hanning-windowed frames with a hop length of 20 ms. Then, 64 log mel-band coefficients are extracted from each frame. A sequence of 300-dimensional frame-wise acoustic embeddings are generated for each audio clip. The final audio embedding is calculated by averaging the frame-wise acoustic embeddings.

3.2. Text Encoder

Word2Vec (Skip-gram model) [21] is utilized as the text encoder to convert textual descriptions into sequences of word embeddings. It is a two-layer fully-connected neural network, which learns word embeddings that are good at predicting surrounding words in a sentence or a document. For the sake of simplicity, we adopt a publicly available pre-trained Word2Vec [22], which is trained on Google News dataset. It consists of 300-dimensional word embeddings for roughly three million case-sensitive English words and phrases. The Word2Vec text encoder converts textual descriptions into sequences of semantic word embeddings word by word. The final caption embedding is computed by averaging the word embeddings.

Rank	Team	**mAP@10** with 95% CI		**R@1** with 95% CI		**R@5** with 95% CI		**R@10** with 95% CI	
1	Xu et al. [10]	0.276	[0.254, 0.299]	0.176	[0.152, 0.200]	0.416	[0.385, 0.447]	0.536	[0.505, 0.567]
2	Mei et al. [11]	0.251	[0.229, 0.273]	0.153	[0.131, 0.175]	0.387	[0.357, 0.417]	0.504	[0.473, 0.535]
3	Lamort et al. [12]	0.221	[0.200, 0.242]	0.131	[0.110, 0.152]	0.343	[0.314, 0.372]	0.466	[0.435, 0.497]
4	Pellegrini [13]	0.216	[0.195, 0.237]	0.127	[0.106, 0.148]	0.321	[0.292, 0.350]	0.463	[0.432, 0.494]
5	Lai et al. [14]	0.215	[0.194, 0.235]	0.122	[0.102, 0.142]	0.328	[0.299, 0.357]	0.478	[0.447, 0.509]
6	Wu et al. [15]	0.188	[0.168, 0.207]	0.107	[0.088, 0.126]	0.303	[0.275, 0.331]	0.413	[0.382, 0.444]
7	Weck et al. [16]	0.128	[0.111, 0.145]	0.077	[0.060, 0.094]	0.188	[0.164, 0.212]	0.284	[0.256, 0.312]
8	Xiao et al. [17]	0.097	[0.083, 0.111]	0.043	[0.030, 0.056]	0.162	[0.139, 0.185]	0.267	[0.240, 0.294]
9	Park et al. [18]	0.075	[0.063, 0.088]	0.033	[0.022, 0.044]	0.127	[0.106, 0.148]	0.208	[0.183, 0.233]
10	Baseline	0.061	[0.049, 0.072]	0.026	[0.016, 0.036]	0.102	[0.083, 0.121]	0.176	[0.152, 0.200]

Table 2: Evaluation results with 95% confidence intervals for the top system of each team.

3.3. Training Objective

The baseline system is trained by optimizing a ranking-based criterion [19], such that audio clips and captions that belong together are more similar in the embedding space than mismatched audio-caption pairs. Specifically, across a batch of N audio-caption pairs $\{(x_n, y_n)\}_{n=1}^{N}$, where y_n is the caption pertaining to an audio clip x_n, we randomly select an imposter clip $\hat{x}_n$ and an imposter caption $\hat{y}_n$ for each audio-caption pair (x_n, y_n). Then, the widely used sampling-based triplet loss [7, 23] is calculated by

$$\begin{aligned} loss = \frac{1}{N}\sum_{n=1}^{N}[&\max(0, S(x_n, \hat{y}_n) - S(x_n, y_n) + 1) \\ &+ \max(0, S(\hat{x}_n, y_n) - S(x_n, y_n) + 1)], \end{aligned} \tag{1}$$

where S is the audio-caption relevance score.

3.4. Baseline Results

The baseline system is trained with batches of 32 audio-caption pairs in the training split for at most 150 epochs, while monitoring the loss (1) on the validation split during the training process. An Adam optimizer with an initial learning rate of 0.001 is adopted to optimize the training process. The learning rate is reduced by a factor of ten once the validation loss does not improve for five epochs. Training is terminated by early stopping with ten epochs.

As shown in Table 3, the baseline system achieves similar performance in terms of mAP@10 and recall scores on the testing split and the evaluation dataset. Specifically, with the evaluation dataset, the theoretical chance levels are $1/1000 = 0.001$ for R@1, $1/200 = 0.005$ for R@5, and $1/100 = 0.01$ for R@10, respectively. In contrast to the theoretical chance levels, the baseline system obtains better recall scores, with an R@1 / R@5 / R@10 of 0.026 / 0.102 / 0.176. The experimental results show that the baseline system can retrieve audio with their corresponding captions, i.e., perform language-based audio retrieval. On the other hand, since the baseline system employs a simple pipeline (e.g., averaging frame-wise acoustic embeddings and word embeddings), the retrieval performance remains limited.

4. CHALLENGE SUBMISSIONS

In this section, we present the evaluation results and analysis of the submissions for language-based audio retrieval.

Dataset	mAP@10	R@1	R@5	R@10
Testing split	0.068	0.032	0.109	0.188
Evaluation	0.061	0.026	0.102	0.176

Table 3: Baseline results on the testing split and the evaluation dataset.

4.1. Evaluation Results

The task received a total number of 31 submissions from nine teams, with maximum four submissions per team allowed. Table 2 shows the results of evaluation metrics (e.g., mAP@10) with 95% confidence intervals (CIs) for the top system of each team, comparing with the baseline system. The 95% confidence intervals for evaluation metrics are calculated using the Jackknife estimate [24].

All the submitted systems outperformed the baseline system in terms of mAP@10 and R@k with $k \in \{1, 5, 10\}$. Particularly, Xu et al. [10] achieved the best performance, with a mAP@10 of 0.276 and a 95% CI between 0.254 and 0.299. Mei et al. [11] ranked second, with their best system having a mAP@10 of 0.251 (95% CI $[0.229, 0.273]$). The top two teams obtained mAP@10 over 0.250 and R@10 over 0.500, in contrast to those of teams ranked third - fifth, having mAP@10 around 0.220 and R@10 around 0.470. Out of the nine teams, six teams achieved mAP@10 over 0.180, R@5 over 0.300, and R@10 over 0.410. The baseline system ranked last with a significantly lower performance.

4.2. Analysis of Submissions

The following analysis is based on the information reported by participating teams.

System summary. All the 31 submitted systems and the baseline system adopted a bi-encoder architecture, which consisted of an audio encoder and a caption encoder, to associate audio with captions. For audio encoders, pre-trained convolutional neural networks (e.g., PANNs [25]) were the most common choice (27 systems from eight teams) among the nine participating teams. For caption encoders, pre-trained Transformer-based language embedding models (e.g., BERT [26] and Sentence-BERT [34]) were frequently employed (28 systems from seven teams). Then, audio and captions were encoded into a common embedding space, where their relevance was scored with cosine similarity (24 systems from six teams) or dot product (seven systems from three teams) of their embeddings. Most of the submitted systems (23 systems from seven

Rank	Team	Audio Modelling	Caption Modelling	Loss Function
1	Xu et al. [10]	PANNs [25]	BERT [26], RoBERTa [27]	InfoNCE loss [28]
2	Mei et al. [11]	PANNs [25]	BERT [26]	NT-Xent loss [29]
3	Lamort et al. [12]	PANNs [25], OpenL3 [30], VGGSound [31], VGGish [32], BART [33]	Sentence-BERT [34]	Triplet loss [6, 7]
4	Pellegrini [13]	PaSST [35]	Sentence-BERT [34]	Triplet loss [6, 7]
5	Lai et al. [14]	ESResNet [36]	Transformer [37]	CE loss [37]
6	Wu et al. [15]	PANNs [25], HTS-AT [38]	Transformer [37]	CE loss [37]
7	Weck et al. [16]	PANNs [25]	RoBERTa [27]	Contrastive loss [39]
8	Xiao et al. [17]	PANNs [25]	Word2Vec [21]	Triplet loss [6, 7]
9	Park et al. [18]	CRNN [20]	Word2Vec [21]	Triplet loss [6, 7]
10	Baseline	CRNN [20]	Word2Vec [21]	Triplet loss [6, 7]

Table 4: Summary of systems with best performance from all teams.

teams and the baseline) were trained by optimizing some contrastive losses (e.g., InfoNCE loss [28] and triplet loss [6, 7]).

Audio encoders. Among the submitted systems, PANNs [25] (e.g., CNN14, Wavegram-Logmel-CNN14 and CNN10) were the most common choice for encoding audio data. There were in total 21 systems from six teams utilizing PANNs as their audio encoders, including the top three teams [10, 11, 12]. Particularly, the CNN14 model was most preferred (20 systems). Xu et al. [10] experimented with multiple pre-trained audio expert models, including CNN14 and Wavegram-Logmel-CNN14. With ensemble of different models, they achieved the best performance. Mei et al. [11] adopted CNN14 as their audio encoder without model ensembles, having their best system ranked second. Tied on third rank, Lamort et al. [12] experimented with aggregating the most number of pre-trained audio expert models (i.e., five models in total, as shown in Table 4). The audio encoders utilized in other systems were PaSST [35], ESResNet [36], HTS-AT [38], and CRNN [18].

Log-mel energies were frequently taken as input among the submitted systems. A total number of 28 systems from eight teams utilized log-mel energies as standalone audio features or in combination with other features (e.g., raw waveform [10]). Other audio features used included log-magnitude spectrogram [12] and log-power spectrogram [14].

Caption encoders. Most of the participating teams (i.e., seven out of nine) preferred Transformer-based language embedding models (e.g., BERT [26]) over word embedding models (e.g., Word2Vec [21]), to learn caption embeddings. As shown in Table 4, pre-trained Transformer-based language embedding models, including BERT [26], RoBERTa [27], and Sentence-BERT [34], were utilized as caption encoders by seven teams (28 systems in total). The other two teams and the baseline converted captions into embeddings with pre-trained Word2Vec word vectors [21]. All teams with Transformer-based models ranked higher than those with pre-trained Word2Vec [21], which showed that Transformer-based models learned caption embeddings more efficiently.

Loss functions. The loss functions utilized in the submissions can be categorized into three groups: contrastive loss (one team), triplet loss (four teams), and N-pair loss (four teams). Contrastive loss [40] takes pairs of samples as input and measures the similarity between two inputs. Triplet loss [6, 7] takes as input triplets of samples, consisting of one anchor sample along with one positive and one negative sample. Compared to contrastive loss, triplet loss focuses on the difference of (dis-)similarities between positive and negative samples to the anchor sample. N-pair loss [40] shares a similar spirit to triplet loss, but extends to multiple negative samples, i.e., one positive and multiple negative samples for an anchor sample. In practice, several versions of N-pair loss were adopted, including InfoNCE loss [28], NT-Xent loss [29], and symmetric cross-entropy (CE) loss [37].

Data augmentation and external data. Data augmentation was adopted by only three participating teams (i.e., Mei et al. [11] on rank two, Lamort et al. [12] on rank three, and Wu et al. [15] on rank six) during their systems' training. In contrast, external audio and textual data was leveraged for optimizing system performance by most participating teams (i.e., seven out of nine). The submitted systems were usually pre-trained with a large amount of external audio and textual data, and then fine-tuned on the task development dataset. For example, the top two teams (i.e., Xu et al. [10] and Mei et al. [11]) pre-trained their systems with AudioCaps [41]. Additionally, Wu et al. [15] trained their systems by involving various external data, including audio samples and text (e.g., labels, file names and captions) from seven external datasets in addition to Clotho v2 [8].

5. CONCLUSIONS

The DCASE 2022 Challenge has introduced the language-based audio retrieval task, which is about using natural language textual captions as queries to retrieve audio signals from a dataset. This paper describes setups of the task, including task description, datasets for system development and evaluation, evaluation metrics, and a task baseline system. Moreover, this paper reports the final evaluation results of the task submissions and their analysis. The evaluation results show that all submitted systems outperformed the baseline in terms of evaluation metrics, with top performance being 0.276 in mAP@10 and 0.536 in R@10.

One immediate observation about the submitted systems is that all of them adopted a bi-encoder architecture consisting of an audio encoder and a caption encoder. Pre-trained convolutional neural networks (e.g., PANNs [25]) were commonly employed as audio encoders, and pre-trained Transformer-based language embedding models (e.g., BERT [26]) were the preferred caption encoders. In addition to the task development dataset, external audio and textual data (e.g., labels and captions) was also frequently leveraged during system training.

6. REFERENCES

[1] J. Dong, X. Li, C. Xu, X. Yang, G. Yang, X. Wang, and M. Wang, "Dual encoding for video retrieval by text," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–10, 2021.

[2] L. Qu, M. Liu, J. Wu, Z. Gao, and L. Nie, "Dynamic modality interaction modeling for image-text retrieval," in *Proc. ACM SIGIR Int. Conf. Res. Dev. Inf. Retr.*, 2021, pp. 1104–1113.

[3] P. Kaur, H. S. Pannu, and A. K. Malhi, "Comparative analysis on cross-modal information retrieval: A review," *Comput. Sci. Rev.*, p. 100336, 2021.

[4] G. Chechik, E. Ie, M. Rehn, S. Bengio, and D. Lyon, "Large-scale content-based audio retrieval from text queries," in *Proc. Int. Conf. Multimed. Inf. Retr.*, 2008, pp. 105–112.

[5] B. Elizalde, S. Zarar, and B. Raj, "Cross modal audio search and retrieval with joint embeddings based on text and audio," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2019, pp. 4095–4099.

[6] A. M. Oncescu, A. Sophia Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2021, pp. 2411–2415.

[7] H. Xie, O. Räsänen, K. Drossos, and T. Virtanen, "Unsupervised audio-caption aligning learns correspondences between individual sound events and textual phrases," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2022, pp. 8867–8871.

[8] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2020, pp. 736–740.

[9] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. Int. Conf. Multimed.*, 2013, pp. 411–412.

[10] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2022 challenge task 6: Audio captioning with audio-text retrieval pre-training," DCASE2022 Challenge, Tech. Rep., 2022.

[11] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Language-based audio retrieval with pre-trained models," DCASE2022 Challenge, Tech. Rep., 2022.

[12] T. Lamort de Gail and D. Kicinski, "Take it easy: Relaxing contrastive ranking loss with CIDEr," DCASE2022 Challenge, Tech. Rep., 2022.

[13] T. Pellegrini, "IRIT-UPS DCASE 2022 language-based audio retrieval system," DCASE2022 Challenge, Tech. Rep., 2022.

[14] Y. Lai, J. Pan, and B. Chen, "A ResNet-Based clip text-to-audio retrieval system for DCASE challenge 2022 task," DCASE2022 Challenge, Tech. Rep., 2022.

[15] Y. Wu, T. Zhang, and K. Chen, "Text-to-audio retrieval via large-scale contrastive training," DCASE2022 Challenge, Tech. Rep., 2022.

[16] B. Weck, M. Pérez Fernández, H. Kirchhoff, and X. Serra, "Aligning audio and text embeddings for the language-based audio retrieval task of the DCASE challenge 2022," DCASE2022 Challenge, Tech. Rep., 2022.

[17] F. Xiao, J. Guan, H. Lan, Q. Zhu, and W. Wang, "Language-based audio retrieval with pretrained CNN and graph attention," DCASE2022 Challenge, Tech. Rep., 2022.

[18] J. Park, C. Hwang, I.-Y. Kwak, and C. Lim, "CAU submission to DCASE 2022 task6b: Language-based audio retrieval using transfer learning," DCASE2022 Challenge, Tech. Rep., 2022.

[19] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 1993, pp. 737–744.

[20] X. Xu, H. Dinkel, M. Wu, and K. Yu, "A crnn-gru based reinforcement learning approach to audio captioning," in *Proc. Detect. Classif. Acoust. Scenes Events Work. (DCASE)*, 2019, pp. 225–229.

[21] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2013.

[22] "Word2Vec," https://code.google.com/archive/p/word2vec, accessed: 2021-09-27.

[23] D. Harwath, A. Torralba, and J. R. Glass, "Unsupervised learning of spoken language with visual context," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1866–1874.

[24] 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 Trans. Audio Speech Lang. Process.*, pp. 992–1006, 2019.

[25] 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 Trans. Audio Speech Lang. Process.*, pp. 2880–2894, 2020.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (NAACL-HLT)*, 2019, pp. 4171–4186.

[27] 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

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

[29] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2020.

[30] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2017.

[31] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman, "VGGsound: A large-scale audio-visual dataset," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2020, pp. 721–725.

[32] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2017, pp. 131–135.

[33] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. Annu. Meet. Assoc. Comput. Linguist. (ACL)*, 2020, pp. 7871–7880.

[34] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Empherical Methods Nat. Lang. Process. (EMNLP)*, 2019, pp. 3982–3992.

[35] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," 2021. [Online]. Available: https://arxiv.org/abs/2110.05069

[36] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "ESResNet: Environmental sound classification based on visual domain models," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 4933–4940.

[37] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 8748–8763.

[38] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2022, pp. 646–650.

[39] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2020, pp. 18 661–18 673.

[40] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 1857–1865.

[41] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. (HLT-NAACL)*, 2019, pp. 119–132.