Audio-Text Cross-Modal Retrieval

Stanford CS224N Custom Project

Vladimir Tourbabin
Department of Computer Science
Stanford University
tourbabv@stanford.edu

Zamir Ben-Hur
Department of Computer Science
Stanford University
zamirbh@stanford.edu

Abstract

Cross-modal retrieval is an important area of research. In particular, the current project focuses on audio clip retrieval using text inquiry. The solution investigated here is based on cross modal metric learning, where two separate encoders for audio and text are learned concurrently by presenting positive and negative examples. The project is focused on investigation of the resulting audio and text embedding spaces. Results indicate that dimension of the embedding spaces may have a significant effect on performance and that higher dimension is not necessarily better. Investigation of uniformity of the text and the audio embeddings clearly shows that the distributions are not uniform. A method to incorporate a non-uniformity penalty into the training pipeline has been proposed, but no benefit has been observed to the performance in the cross-modal retrieval task.

1 Key Information to include

External collaborators - none, External mentor - none, Sharing project - N/A.

2 Introduction

Cross-modal retrieval tasks gained extensive attention in recent years and have made great progress with the advancement of deep learning technologies [1, 2]. Audio-text retrieval is the task of retrieving a desired audio clip or caption from a database of candidates given a query in the other modality. This task can be useful in applications such as clip search, audio book production and movies. Audio-text retrieval is a challenging task as it requires to learn a robust feature representations for both the acoustic and textual modalities. Additionally, the task requires to capture the fine-grained interaction between the learned acoustic and textual features and aligning them in a shared embedding space.

Early works for audio-text retrieval focused on tag-based audio retrieval, where the queries were words as opposed to full sentences [3, 4]. More recent work [5, 6] started exploring free-form language-based audio retrieval, which is also the focus of the current project. See Section 3 for a more detailed discussion.

Audio-text retrieval can be implemented using two independent networks, one for audio encoder and the second for the text encoder. The goal of these two networks is to encode the audio and text into a shared embedding space, where positive pair of audio-text will be closer than negative pairs. The training objective of such model is consistent with that of metric learning, which has been a popular choice for the optimization of the cross-modal retrieval models [7]. Recently, Mei *et al.*[6] studied different ways to learn the audio and text cross-modal embedding space by comparing various training loss functions. The comparison was carried out by directly evaluating the performance on the downstream audio clip retrieval task. However, little attempt is made to gain a deeper insight into the learned cross-modal spaces and their properties. Having such a deeper understanding can lead to an additional perspective on the limitations of the current methods and potentially suggest

a better approach. In this project we aimed to address this gap by analyze the learned cross-modal representation. In particular, we investigated the two following aspects:

1. Dimension of the embedding spaces and its relation to the performance in the downstream task of cross-modal retrieval.
2. Uniformity of the resulting text and audio embedding distributions. Uniformity may shed light into the extent to which all of the available space is being utilized and how independent the specific features being learned. It is hypothesized that enforcing uniformity during training may be beneficial for various aspects of the final solution including performance in the downstream or convergence during training.

The current report first outlines in more detail the approach to the cross-modal retrieval task including the pre-trained models, data, and metrics. Then, we dive deeper into the two experiments carried out as part of this project. In the first experiment we familiarize ourselves with the training pipeline proposed in [6] and learn how to modify it by investigating the effect of the embedding space dimension on its performance. In the second experiment, we carry out an analysis of uniformity of the resulting text and audio embedding spaces and propose a modification of the training pipeline to incorporate a non-uniformity penalty. Both experimental sections include details relevant to the specific experiment, results, and a discussion. A brief conclusion and future work completes the report.

3 Related Work

The problem of cross-modal audio-text retrieval was first tackled at scale by Chechik *et al.* [3]. In this paper a combined scoring function has been exploited for the retrieval of an audio clip from its associated text label. The main limitation of this work is that the query must be exactly as it appears in the index. A more flexible approach was proposed by Slaney [8], where new sounds could be associated with existing labels. However, a hierarchical language model was used, which, according to the authors, limits the scalability of the model. A more recent work [4] suggested to find a lexico-acoustic spaces in a data-driven way, which is more robust and scalable. All of the work mentioned so far is based on words as queries instead of a free-form language, which is more natural for the downstream audio-text retrieval task. Lately, Koepke *et al.* [5] established the first benchmark for free-form language-based audio retrieval. In this paper, pre-trained models and common ideas from video retrieval have been adopted to address the scarcity of the audio-text data. Finally, Mei *et al.*[6] presented a full free-form language-based audio-text retrieval model and studied various learning metrics for the training of the model. It has been concluded that the learning metric have a significant effect on the downstream task performance, where the NT-Xent loss [9] (detailed below) outperformed the triplet-based losses [10] and showed stable performances with respect to various training settings and datasets.

4 Approach

The audio and the text encoders proposed in [6] were used. The text encoder comprises the BERT model [11] and an additional multilayer perceptron (MLP). The MLP consists of two linear layers with a ReLU activation in between. A "<CLS>" token is appended at the start of each sentence. For the audio encoder, a PANNs (Pre-trained audio neural networks) model has been employed [12] (based on the ResNet-38), which is a pre-trained model on audio tagging task that showed to provide a robust audio representation and high performance on various audio related tasks. A max pooling layer and an MLP similar to the text encoder was added to the network. Log mel-spectrograms are used as the audio features, which are extracted using a 1024-points Hanning window with 320-points hop size and 64 mel bins. Figure 1 shows the system diagram.

The output of the audio and text encoders are compared using a cosine similarity metric:

$$s_{ij} = \frac{f(a_i) \cdot g(t_j)}{||f(a_i)||_2 ||g(t_j)||_2}, \tag{1}$$

where a_i and t_i are the audio and text pair, such that (a_i, t_i) is a positive pair while $(a_i, t_{j,j\neq i})$ is a negative pair. f and g are the audio and text encoders, respectively. Then, a loss can be defined

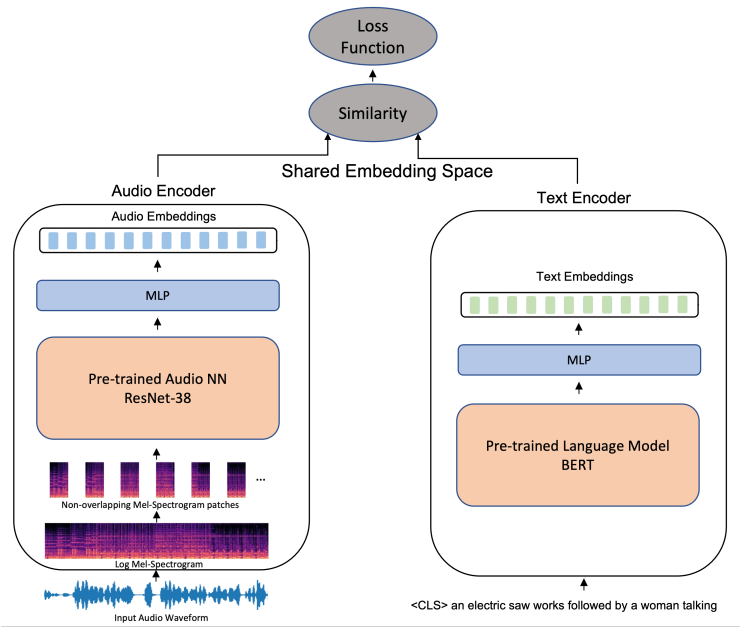

Figure 1: Diagram of the training system, where the goal is to find the optimal shared embedding space

such that the similarity score of positive pairs will be higher than that of negative pairs. The NT-Xent loss [9] has been chosen, as it perform the best in [6]. This is a contrastive loss based on softmax:

$$\mathcal{L} = -\frac{1}{B}\left(\sum_{i=1}^{B}\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{B}\exp(s_{ij}/\tau)} + \sum_{i=1}^{B}\log\frac{\exp(s_{ii}/\tau)}{\sum_{j=1}^{B}\exp(s_{ji}/\tau)}\right), \tag{2}$$

where B is the batch size and τ is a temperature hyper-parameter.

4.1 Data

We used the AudioCaps dataset [13]. The dataset contains $50k$ audio clips of 10 sec length with paired human-annotated text captions. All audio clips are trimmed to the same length of 10 seconds. The dataset was splitted to 49,274 audio clips for training, 494 for validation and 957 for test.

4.2 Evaluation

Recall at rank k (R@k) is used as the evaluation metric, which is the popular cross-modal retrieval evaluation protocol. R@k measures the percentage of targets retrieved within the top k ranked results, thus the higher the score, the better the performance. We report R@1, R@5, and R@10.

5 Experiment #1 - embedding space dimension

The goal of this experiment was to evaluate the effect of the shared embedding space dimension on the performance of the audio-text retrieval task. Two complementary sub-experiments were conducted in this experiment:

5.1 Experiment 1.a

All the models were trained for 50 epochs using Adam optimizer. The learning rate is set to 10^{-4} and is decayed to 1/10 of itself every 20 epochs. The batch size is set to 64. The temperature hyper-parameter τ is set to 0.07. The best model is selected based on the sum of recalls on the validation set. All experiments are carried out on an AWS server. Figure 2 shows an example of

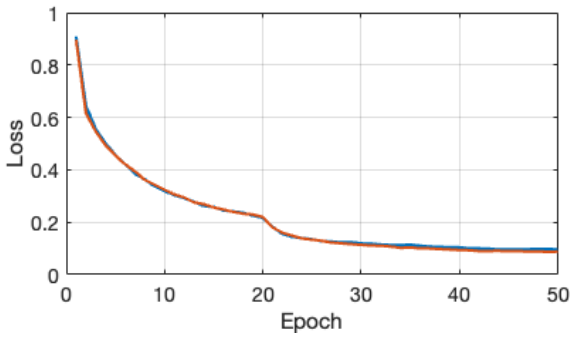

Figure 2: Example of the loss convergence as a function of epoch.

the loss convergence as a function of epoch. It can be seen that the model is able to converge in 50 epochs. A significant increase in convergence rate after 20 epoch is probably due to the learning rate update by a factor of 1/10. The same update at epoch 40 does not seem to further benefit the training.

Exp. 1.a aimed to evaluate the effect of the embedding space dimension. To save computation time, freezed pre-trained encoding models were used (fine-tune the models takes 30min per epoch, i.e. 24hr for training, while for freezed models third of time is needed). The size of the shared embedding space has been varied between 128, 256, 512, 1024 and 2048, while the original paper reported only 1024. The learned acoustic semantic embeddings were normalized.

5.2 Experiment 1.b

The aim of this sub-experiment was to complement Exp. 1.a by investigating performance with fine-tuning. This was carried out only for a single condition to save computation time, while allowing for comparison to the current state-of-the-art results published in [6]. Similar parameters to Exp 1.a were used, with the exception of the batch size of 32 to match the best results in the original paper.

5.3 Results and discussion

The results of Exp. 1.a are presented in Tab. 1. The table shows the R@k for the various embeddings sizes, for both text-to-audio and audio-to-text retrieval tasks. It can be seen that using embedding vector of 512 is consistently better (beside R@10 for Audio-to-Text). This result is surprising since the original paper used the 1024 size.

For a better comparison with the original paper, Tab. 2 shows the results of Exp. 1.b in comparison to the original paper results with embedding size 1024. It can be seen that our suggested model (with size of 512) perform better for all R@k.

These results clearly indicate two important findings: a) that embedding dimension is an important hyper parameter to be chose carefully and b) larger embedding dimension is not necessarily better.

Table 1: Cross-modal retrieval performance as a function of embedding space dimension using pre-trained and frozen encoders.

Embeddings Size	**Text-to-Audio**			**Audio-to-Text**		
	R@1	R@5	R@10	R@1	R@5	R@10
128	16.1	45.9	63.2	17.0	49.4	64.5
256	16.7	46.5	62.9	17.0	48.9	64.8
512	**17.2**	**47.9**	**64.2**	**20.9**	**50.5**	67.2
1024	16.6	46.5	62.6	17.9	49.6	**68.3**
2048	16.8	46.5	63.7	19.8	49.9	67.2

Table 2: Cross-modal retrieval performance comparing as a function of embedding space dimension using pre-trained and fine-tuned encoders.

Embeddings Size	Text-to-Audio			Audio-to-Text		
	R@1	R@5	R@10	R@1	R@5	R@10
512	**34.4**	**69.8**	**82.9**	**41.9**	**74.0**	**84.2**
1024 (From [6])	33.9	69.7	82.6	39.4	72.0	83.9

6 Experiment #2 - embedding space uniformity

6.1 Uniformity analysis

In order to analyze the uniformity of the resulting text and audio embedding spaces, two alternative metric have been suggested: point to point uniformity and voxel-based uniformity. In particular, point to point uniformity has been adopted from [14]. Given a set of N points $X = \{x_i\}_i^N$, point-to-point uniformity, σ, can be computed in two steps. First, for each point in the set obtain the distance to its closest neighbour:

$$\lambda_i = \min_{j \neq i} \|x_i - x_j\|. \tag{3}$$

Second, compute the uniformity as the standard deviation of the distances normalized by their mean:

$$\sigma = \frac{1}{\bar{\lambda}} \left(\frac{1}{N} \sum_i (\lambda_i - \bar{\lambda})^2 \right)^{0.5}, \tag{4}$$

where $\bar{\lambda} = \frac{1}{N} \sum_i \lambda_i$. The lower the value of this metric, the more uniform the set X is believed to be. In the extreme case where all λ_i are equal, the metric attains its lowest value $\sigma = 1$.

The second metric, voxel-based uniformity, has been proposed as part of this project. It is defined by first dividing the volume of interest, V, into a number of equal sub-volumes, $\{V_i\}_{i=1}^I$ called voxels, such that $\cup_i V_i = V$. Then, a histogram $\{h_i\}_{i=1}^I$ is computed by counting the number of points in X that belong to each voxel. Finally, the voxel-based uniformity h is computed as the entropy of the normalized histogram:

$$h = entropy\left(softmax(\{h_i\}_i)\right) \tag{5}$$

Note that, as opposed to the point to point uniformity, the voxel-based uniformity is expected to be larger for more uniform set attaining its maximum value of $\log I$ in the extreme case where all h_i are equal.

In this project, the uniformity was computed using normalized embeddings. Note that applying the voxel-based uniformity to a high-dimensional space can be computationally demanding. To limit the amount of computations needed to calculate h, we have divided the space into only two parts, positive and negative, along each dimension, and randomly chose only 8 dimensions at a time. This resulted in only 256 voxels. The calculation was repeated for a 100 times and averaged.

Uniformity of the text and audio embedding distributions has been analyzed using the above two metrics. The analysis was carried out as a function of the embedding space dimension. We have also measured the uniformity of a uniformly distributed set and of a clustered set to obtain a better intuition of the two metric values. Example plots of the different distributions are presented in Fig. 3. The corresponding uniformity values are summarised in Table 3.

From the example distributions in Fig. 3 and from the numbers in Table 3, it is evident the the resulting embedding distributions are not quite uniform. Instead, they fall somewhere in between the uniformly distributed set and the mixture of Gaussains set. Degree of non-uniformity seems to vary depending on the specific metric. Another important observations is that the uniformity does not seem to depend on the embedding space dimension.

6.2 Modified training pipeline

Inspired by this result above, it was proposed to incorporate uniformity into the loss function during training to explore potential benefits of of doing so on various aspects of the solution.

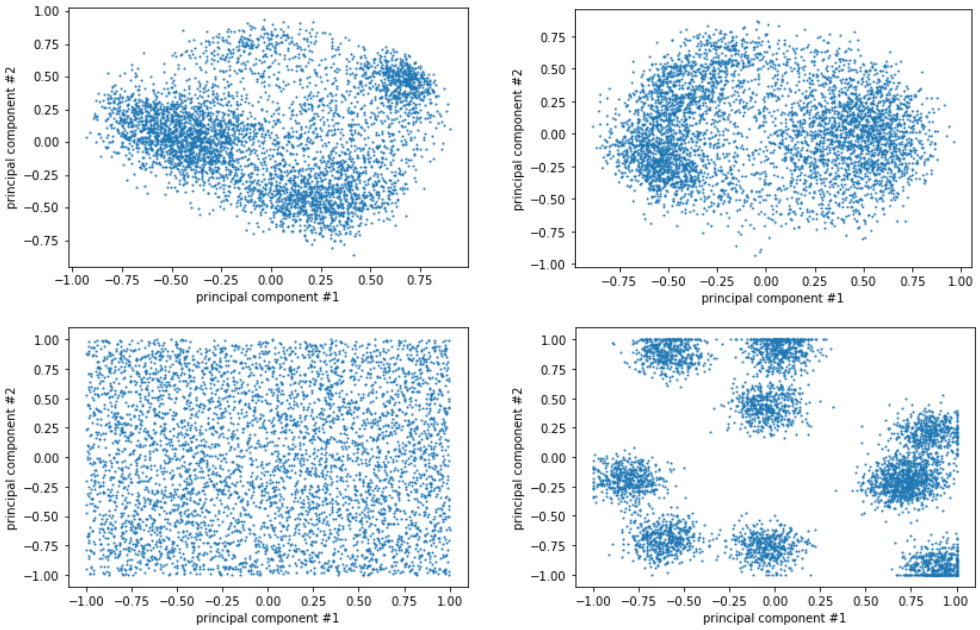

Figure 3: Example plots of the four different distributions. To produce these plots, the actual set has been analyzed using PCA and only the first two principal components are plotted here. The distributions are (a) text (caption) embeddings, (b) audio embeddings, (c) set drawn from a uniform distribution, (d) set drawn from a mixture of 10 Gaussians with equal variance of $1/10$ and uniformly distributed random centroids.

Table 3: Uniformity analysis of the resulting text and audio embeddings along with a uniformly distributed and a Gaussian mixture set as a function of the embedding space dimension. See text for a definition of the uniformity metrics σ and h.

embd	**point to point uniformity σ**				**voxel-based uniformity h**			
dimension	**text**	**audio**	**uniform**	**Gaussians**	**text**	**audio**	**uniform**	**Gaussians**
128	0.27	0.29	0.19	0.35	3.96	3.98	4.13	2.79
256	0.28	0.29	0.19	0.35	3.94	3.99	4.12	2.75
512	0.28	0.28	0.19	0.34	3.96	4.01	4.12	2.82
1024	0.28	0.27	0.19	0.35	3.98	4.00	4.13	2.83
2048	0.27	0.28	0.19	0.34	3.97	4.00	4.13	2.81

In order to be included into the training loss, the uniformity calculation has to be differentiable. To that end we chose to work with voxel-based uniformity and used a pretrained fully connected network to predict the voxel to which each embedding belons. Then, summing across the batch dimension results in an estimated histogram, $\{h_i\}_{i=1}^{I}$, which is used to compute the batch uniformity. The network architecture is shown in Fig. 4. In order to train the network, 100k examples of 8-dimensional vectors were drawn from uniform distribution and assigned into one of the 256 voxels (see code for additional details). The dataset has been split 90k|10k for training and validation, respectively. The network was trained using Adam optimizer for 100 epochs and a small weight decay of $4e-4$, which was shown to effectively prevent over-fitting. The resulting network has been tested using an additional 10k dataset that has been held back resulting in 97.0% accuracy.

This voxel predicting network has been incorporated into the modified training pipeline shown in Fig. 5. The pipeline allow to incorporate a non-uniformity penalty by subtracting the uniformity value obtained for each batch. Uniformity of both, text and audio encoders contribute symmetrically to the combined loss. The level to which the the uniformity affect the loss is controlled through the significance parameter α, which is also symmetrically applied to the two modalities. In each batch, 8 of the embeddings features are selected at random using uniform distribution, hence only

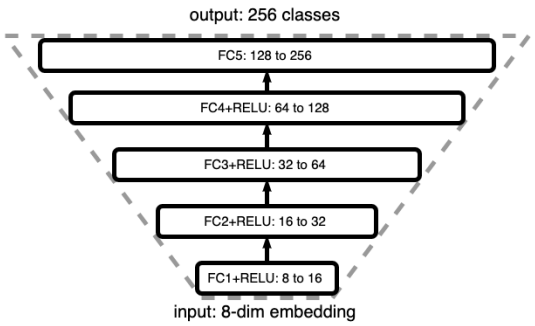

Figure 4: Architecture of the fully-connected NN designed to predict to which out of the 256 voxels each embedding belongs.

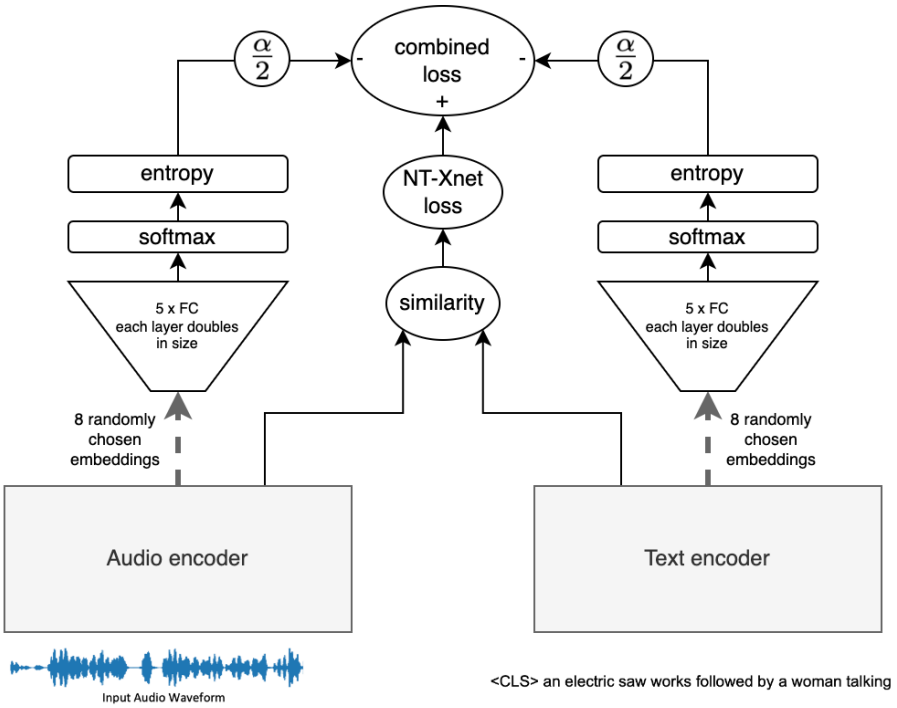

Figure 5: Modified training pipeline used to include uniformity into the loss function during training. A non-uniformity penalty is added to the combined loss by subtracting uniformity of both text and audio embedding batches symmetrically.

8 dimensions are updated with the uniformity gradients. It is believed that the randomization will eventually lead to updating all of the embedding dimensions.

6.3 Results and discussion

The effect of the uniformity loss component on the downstream task performance has been investigated. The uniformity of the resulting embedding distributions and the cross-modal retrieval performance, $R@k$, are summarized in Tab. 4. It can be seen that the cross-modal retrieval does not seem to benefit from the non-uniformity penalty; on the contrary, a consistent trend of reduced $R@k$ values is obtained when increasing the uniformity significance parameter α. Additional investigation with varying hyper parameters and potentially different ways of incorporating the non-uniformity penalty might be needed to fully uncover the potential benefits.

Table 4: Cross-modal retrieval performance and the embedding space uniformity as a function of the significance parameter α.

α	**Text-to-Audio**			**Audio-to-Text**			**Uniformity**	
	R@1	R@5	R@10	R@1	R@5	R@10	**Audio**	**Text**
0	**34.4**	**69.8**	**82.9**	**41.9**	**74.0**	**84.2**	**4.02**	**3.98**
0.10	32.3	68.3	82.1	37.8	72.1	83.6	3.99	3.97
0.33	31.6	66.4	80.0	37.8	70.4	82.1	3.97	3.95
1.00	26.6	63.3	78.4	30.6	62.5	80.6	4.00	3.95

7 Conclusion

This project presents a comprehensive analysis of the shared embedding space of the cross-modal audio-text retrieval model. Investigation of both the dimensionality and uniformity of the embedding space has been performed. It has been shown that the dimensionality of the shared embedding space is an important parameter of the model, which should be carefully selected. In our experiments the optimal embedding space size that outperforms state-of-the-art was 512 for both frozen and fine-tuned pre-trained models. Voxel-based uniformity metric was proposed and incorporated into the training pipeline. Initial investigation of its effect on the system performance are inconclusive. No benefit has been observed so far to performance in the downstream task of cross-modal retrieval. Future work may include additional experimentation with different ways to incorporate the non-uniformity penalty and further hyper parameter tuning.

References

[1] Kunpeng Li, Yulun Zhang, Kai Li, Yuanyuan Li, and Yun Fu. Visual semantic reasoning for image-text matching. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 4654–4662, 2019.

[2] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 214–229. Springer, 2020.

[3] Gal Chechik, Eugene Ie, Martin Rehn, Samy Bengio, and Dick Lyon. Large-scale content-based audio retrieval from text queries. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 105–112, 2008.

[4] Benjamin Elizalde, Shuayb Zarar, and Bhiksha Raj. Cross modal audio search and retrieval with joint embeddings based on text and audio. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4095–4099. IEEE, 2019.

[5] A Sophia Koepke, Andreea-Maria Oncescu, Joao Henriques, Zeynep Akata, and Samuel Albanie. Audio retrieval with natural language queries: A benchmark study. *IEEE Transactions on Multimedia*, 2022.

[6] Xinhao Mei, Xubo Liu, Jianyuan Sun, Mark Plumbley, and Wenwu Wang. On Metric Learning for Audio-Text Cross-Modal Retrieval. In *Proc. Interspeech 2022*, pages 4142–4146, 2022.

[7] Kevin Musgrave, Serge Belongie, and Ser-Nam Lim. A metric learning reality check. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 681–699. Springer, 2020.

[8] Malcolm Slaney. Semantic-audio retrieval. In *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–4108. IEEE, 2002.

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[10] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[12] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition, 2019.

[13] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. AudioCaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[14] Max Gunzburger and John Burkardt. Uniformity measures for point samples in hypercubes, 2004.