Killing two birds with one stone: can an audio captioning system also be used for audio-test retrieval?

Etienne Labbé, Thomas Pellegrini, Julien Pinquier

KILLING TWO BIRDS WITH ONE STONE: CAN AN AUDIO CAPTIONING SYSTEM ALSO BE USED FOR AUDIO-TEXT RETRIEVAL?

Étienne Labbé[1], *Thomas Pellegrini*[1,2], *Julien Pinquier*[1]

[1]IRIT, Université Paul Sabatier, CNRS, Toulouse, France
[2]Artificial and Natural Intelligence Toulouse Institute (ANITI)
{etienne.labbe,thomas.pellegrini,julien.pinquier}@irit.fr

ABSTRACT

Automated Audio Captioning (AAC) aims to develop systems capable of describing an audio recording using a textual sentence. In contrast, Audio-Text Retrieval (ATR) systems seek to find the best matching audio recording(s) for a given textual query (Text-to-Audio) or vice versa (Audio-to-Text). These tasks require different types of systems: AAC employs a sequence-to-sequence model, while ATR utilizes a ranking model that compares audio and text representations within a shared projection subspace. However, this work investigates the relationship between AAC and ATR by exploring the ATR capabilities of an unmodified AAC system, without fine-tuning for the new task. Our AAC system consists of an audio encoder (ConvNeXt-Tiny) trained on AudioSet for audio tagging, and a transformer decoder responsible for generating sentences. For AAC, it achieves a high SPIDEr-FL score of 0.298 on Clotho and 0.472 on AudioCaps on average. For ATR, we propose using the standard Cross-Entropy loss values obtained for any audio/caption pair. Experimental results on the Clotho and AudioCaps datasets demonstrate decent recall values using this simple approach. For instance, we obtained a Text-to-Audio R@1 value of 0.382 for AudioCaps, which is above the current state-of-the-art method without external data. Interestingly, we observe that normalizing the loss values was necessary for Audio-to-Text retrieval.

Index Terms— Automated audio captioning, audio-text retrieval, ConvNeXt, DCASE Workshop

1. INTRODUCTION

In recent years, audio-language tasks have received greater attention due to advances in machine learning for text processing. For example, the Automated Audio Captioning (AAC) task aims to create machine learning systems that produce a sentence describing an audio file, while the Audio-Text Retrieval (ATR) task aims to use a caption to extract an audio from its database (Text-to-Audio, T2A) or use an audio to retrieve a caption its database (Audio-to-Text, A2T). Research on these tasks is also boosted by the DCASE Challenge and Workshop[1], which proposed two tasks dedicated to AAC and T2A. Although these tasks appear to be closely related, they are usually performed by two different systems and architectures. Those systems can sometimes share common weights [1], but they need to be trained differently on several phases. In the image captioning task, the authors of [2] proposed to use a captioning system by describing each image and compare these descriptions to the captions instead of the images. In this paper, we propose another method for using an AAC system to perform the ATR task, and we investigate the implications of using this system in this way.

[1]https://dcase.community/

2. SYSTEM DESCRIPTION

2.1. AAC system architecture

To achieve the AAC task, we employ deep neural network with an encoder-decoder architecture. We trained a ConvNeXt [3] (CNext) model for audio tagging and used it as an encoder to produce frame-level features to overcome the limitation of audio-language data. The ConvNeXt was trained on AudioSet [4] audio tagging dataset without the AudioCaps [5] audio captioning dataset files to avoid biases. This encoder achieves a high mAP score of 0.462 on AudioSet. The details of the architecture and training hyperparameters are given in [6]. The encoder gives a list of features of shape 768×31 for a 10-seconds audio clip, which are projected by a sequence of dropout set to 0.5, dense layer, a ReLU activation and another dropout set to 0.5. The decoder is a standard transformer decoder architecture [7] with six decoder layers blocks, four attention heads per block, a feedforward dimension of 2048, a GELU [8] activation function and a global dropout set to 0.2. Unlike a lot of AAC and ATR systems, no pre-trained weight has been used for the decoder/word modelling part. We found that freezing the ConvNeXt encoder leads to lower variances, so we decided to pre-compute all its embeddings to train only the decoder part. The whole model contains 28M frozen parameters and 12M trainable parameters.

2.2. Data augmentation

During our training with the decoders, we added three different augmentations on audio and input word embeddings to reduce overfitting and improve model generalization. mixup [9] modifies the input audio and words embeddings during training, with α set to 0.4. Each embedding is mixed with another one in the current batch, except for the target, which remains unmixed. Label Smoothing [10] is applied to the targets one-hot vectors to reduce the maximal probability of each word and limit the confidence of the model. Finally, SpecAugment [11] masks a part of the audio frame embeddings, with 6 stripes dropped with a maximal size of 4 in time axis and 2 stripes dropped with a maximal size of 2 in feature axis.

2.3. Using a captioning system for retrieval

The first idea to use an AAC system for ATR is to generate predictions to describe each audio file and compare each text query to each description using a metric like BLEU, CIDEr-D or SBERT, as proposed in [2], but we found low results using this strategy. We

Table 1: AAC results on Clotho and AudioCaps testing subsets. Ours results are averaged over 5 seeds. WC stands for WavCaps [12] dataset. Best values for each dataset/metric are in **bold**, and best values without external data are underlined.

Dataset	System	Train data	#params	METEOR	CIDEr-D	SPICE	SPIDEr	SPIDEr-FL
CL	BEATs+Conformer [13]	CL+AC	127M	**.193**	**.506**	**.146**	**.326**	**.326**
	CNN14-trans [14]	CL	88M	.177	.441	.128	.285	N/A
	CNext-trans (ours)	CL	40M	.189	.464	.136	.300	.298
AC	HTSAT-BART [12]	AC+WC	171M	**.250**	**.787**	.182	**.485**	N/A
	Multi-TTA [15]	AC	108M	.242	.769	.181	.475	N/A
	CNext-trans (ours)	AC	40M	.246	.763	**.183**	.473	.472

believe that AAC systems tend to produce less detailed and diversified sentences than references, which leads to a loss of information when using it to summarize the audio content into a single sentence. Typically, the vocabulary size used during inference is only around 617 distinct words over the 1839 words present on average in the references for the Clotho development-testing subset. AAC systems are usually trained to predict the next token of a sentence using previous words and the audio file. This means that the model actually takes as input an audio and a caption, and the loss could be used to score this input. We decided to simply use the Cross-Entropy (CE) loss used in training to score each pair, and expecting that an AAC system should be able to give a higher loss value when the input caption does not match the input audio file than when they match. Equations 1a and 1b describe how an audio and text element are retrieved using the CE.

$$\text{T2A}(t, A, f) = \underset{a \in A}{\operatorname{argmin}} \, \text{CE}(f(a, t_{\text{prev}}), t_{\text{next}}) \tag{1a}$$

$$\text{A2T}(a, T, f) = \underset{t \in T}{\operatorname{argmin}} \, \text{CE}(f(a, t_{\text{prev}}), t_{\text{next}}) \tag{1b}$$

where t corresponds to a caption, T is the list of all captions, a is an audio file from the A list of audio files. f is the AAC system which produces the distributions of probabilities for the next words t_{next} given the previous words t_{prev} in the context of an audio file.

3. EXPERIMENTAL SETUP

3.1. Datasets

AudioSet [4] is the largest audio tagging dataset publicly available and contains 2M pairs of audio/tag. The audio files last for 10 seconds extracted from YouTube videos and the dataset contains 527 different sound events tags. Clotho [16] (CL) is an AAC dataset containing 6974 audio files ranging from 15 to 30 seconds in length extracted from the FreeSound website. The dataset is divided into three splits used respectively for training, validation and testing, containing five captions per audio file. In our experiments, each audio file is resampled from 44.1 kHz to 32 kHz. During training, we randomly select one of five captions for each audio file. AudioCaps [5] (AC) is the largest AAC dataset written only by humans, containing 51308 audio files from the AudioSet dataset. Since original YouTube videos are removed or unavailable for various reasons, our version of the train split contains 46230 out of 49838 files, 464 out of 495 in the validation split and 912 out of 975 files in the test split. In addition, we slightly improve caption correctness in the training subset by manually fixing 996 invalid captions with grammatical and typographic errors. For the two AAC datasets, captions are put in lowercase and all punctuation characters are removed. The codebase used to download, read and extract data is a package named aac-datasets[2].

3.2. Metrics

For the AAC task, we report the five metrics used in the DCASE Challenge task 6a. METEOR [17] is based on the precision and recall of the words. CIDEr-D [18] uses the TF-IDF scores of the shared n-grams between candidates and references. SPICE [19] builds a graph representing the scene described by the captions and compute an F-score with its common edges. SPIDEr [20] average the two previous metrics and finally, SPIDEr-FL[3] is a combination of the SPIDEr metric with a pre-trained system designed to detect fluency errors. When one of them is detected, the SPIDEr score is divided by a factor of 10. The codebase for AAC metrics is available as a public Pip package[4] named aac-metrics. For the ATR task, we use the Recall@k metric, which measures if a relevant (ground truth) element is in the top-k retrieved elements.

3.3. Hyperparameters

The number of training epochs K set to 400 with a batch size set to 512. The optimizer used is AdamW with an initial learning rate (lr_0) set to $5 \cdot 10^{-4}$, β_1 set to 0.9, β_2 set to 0.999, ϵ set to 10^{-8} and weight decay set to 2. Weight decay is not applied to the bias contained in the network. The learning rate is decreased during training at the end of each epoch k using a cosine scheduler rule: $\text{lr}_k = \frac{1}{2}\big(1 + \cos(\frac{k\pi}{K})\big)\text{lr}_0$. The gradient L_2-norm is clipped to 1 to avoid collapsing across seeds, the label smoothing reduces maximal target probability by 0.2 and the mixup α hyperparameter is set to 0.4. Since only the projection and the decoder part are trained, a single AAC experiment runs in one hour on AC and three hours on CL datasets with one V100 graphics card. To validate our model, we used the FENSE metric [27] which is based on computes the cosine similarity of the embeddings produced by a pre-trained Sentence-BERT model combined with the same fluency error detector used in SPIDEr-FL. During validation and inference, we used the standard beam search algorithm to generate better sentences. In order to limit the number of repetition tokens, we forced the model to avoid generating the same word twice in a single sentence, except for stop words defined in NLTK package.

[2] https://pypi.org/project/aac-datasets/0.3.3/

[3] https://dcase.community/challenge2023/task-automated-audio-captioning

[4] https://pypi.org/project/aac-metrics/0.4.2/

Table 2: Audio-language retrieval results on Clotho and AudioCaps testing subsets. Ours results are averaged over five seeds. WC stands for WavCaps dataset. Best values for each dataset/task/metric are in **bold**, and best values without external data are underlined. The asterisk * denotes the results scaled by a min-max strategy described in 4.3.

Retrieval dataset	System	Training dataset(s)	#params	Text-to-audio			Audio-to-text		
				R@1	R@5	R@10	R@1	R@5	R@10
CL	PaSST-N[4] [21]	CL+AC+WC	441M	**.261**	**.553**	**.693**	N/A	N/A	N/A
	CNN14-BERT [12]	CL+WC	214M	.215	.479	.663	**.271**	**.527**	**.663**
	CNN14-BERT [22]	CL	192M	.167	.410	.539	N/A	N/A	N/A
	Triplet-weighted [23]	CL	185M	.142	.366	.497	.169	.381	.514
	TAP+PMR [24]	CL	185M	.171	.396	N/A	.182	.399	N/A
	CNext-trans (ours)	CL	40M	.137	.349	.480	.148*	.404*	.541*
AC	HTSAT-BERT [12]	AC+WC	141M	.422	.765	.871	**.546**	**.852**	**.924**
	ONE-PEACE [25]	CL+AC+7 others	2B	**.425**	**.775**	**.884**	.510	.819	.920
	MMT [26]	AC	290M	.361	.720	.845	.396	.768	.867
	Multi-TTA [15]	AC	187M	.347	.703	.832	.402	.740	.872
	TAP+PMR [24]	AC	185M	.368	.727	N/A	.417	.762	N/A
	CNN14+TAP+PMR [24]	AC	192M	.334	.688	N/A	.431	.733	N/A
	CNext-trans (ours)	AC	40M	.382	.733	.853	.398*	.814*	.919*

4. RESULTS

4.1. AAC results

The AAC results are given in Table 1. We also reported the SOTA scores for each dataset, without reinforcement learning, without ensemble method and with or without external captioning datasets. On CL, our model performs better than the previous SOTA without external data (CNN14-trans) in all metrics and uses more than twice fewer parameters (40M instead of 88M). We believe this is mainly due to our stronger pretrained encoder, which has a higher mAP score on AudioSet and produces better features for AAC. On AC, the model reach a score very close to the Multi-TTA method, with only 0.002 absolute difference in SPIDEr despite having an unbiased encoder not trained on the testing files of AC.

4.2. ATR results

Retrieval results are shown in Table 2. Just as AAC results, we reported the SOTA methods without ensemble methods and with or without external captioning datasets. Since all values are not always reported, we added several SOTA methods to compare our system with at least one other methods for each column. For the T2A task on the CL dataset, our model performs better than the DCASE baseline, but worse than most SOTA methods. However, the system achieves the highest scores on AudioCaps without external data. Somewhat surprisingly, our system outperforms other methods without external data on the A2T task on R@5 and R@10, but not on the R@1 metric on both datasets.

4.3. Why A2T performance is so low in the first setting?

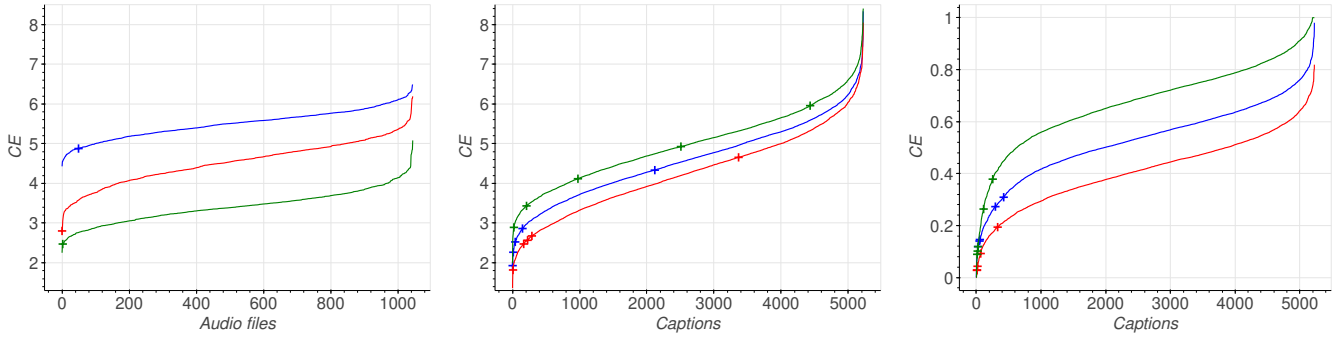
We found that even if our system performs well on T2A task, the results on A2T one were really low compared to the SOTA ones. The system reaches an R@1 of 0.146 on AC and 0.038 on CL when using raw loss values. We found that this is caused by a subset of the captions, where the loss values are almost always lower than the others for all audio files. We can see in Figure 1a that the curve of the bottom corresponding to the losses of one caption with all other audio files that the model considers this caption as much more likely than the ones on the top of the figures, a so for most audio files. In particular, only 120 unique captions are retrieved for 1045 queries during the A2T task with raw losses, but we did not find a strong correlation between these captions and the frequencies of their words or their length. In order to clarify why it impacts only the A2T task and not T2A, we provide a simple example in Table 3. This example shows the loss values for three different audio A_i with their corresponding captions C_i. When we perform T2A task, we select the retrieved audio A_i with the lowest loss value in the column i, which reach a perfect score in that case. However, when we perform A2T, only the caption C_1 is retrieved, because its column has a range of value different from the others, which explains the poor results when using raw loss values. To tackle this problem, we propose a post-processing which scales each "column" (i.e. each series of values corresponding to a single caption). In particular, we tried to normalize, standardize, but a simple min-max scaling has led to the best results. We also added a rule when two retrieved captions has the same score (zero when they are the minimal value of their column) by using their original losses to decide which one will be used. The impact of this scaling on the A2T losses are given in figures 1b and 1c.

Table 3: Real loss values over 3 audio files and captions.

	C_1	C_2	C_3
A_1	**1.7**	8.4	8.1
A_2	2.1	**7.6**	8.5
A_3	2.0	8.3	**6.5**

5. BENEFITS AND DOWNSIDES OF USING AAC SYSTEM

Recently, the authors of this paper [28] showed that ATR systems usually fails to capture high-level relations between sounds by showing corrupted captions to an ATR system. More precisely, they propose to replace in caption the word "*after*" by "*before*" and vice versa to invert the sequence of sound events described and name



(a) Sorted losses for 3 captions over audios. (b) Sorted losses for 3 audios over captions. (c) **Scaled** sorted losses for 3 audios over captions.

Figure 1: Losses for 3 queries over all retrieved items. The position of the relevant (ground truth) elements are shown with a cross.

this the Before-After Test (BAT). The ATR system should be able to give a lower score for an incorrect input caption than for a correct one. We believe that audio-language systems should be able to capture that kind of information than audio event classes, but the actual metrics do not usually reflect the model performance on it. In addition to the perturbation proposed by them, we proposed to switch the relation type from sequence to superposition and vice versa by replacing some words or inverting the propositions of the sentence. For example, the sentence "*a man speaks* ***then*** *a dog barks*" can become "*a man speaks* ***as*** *a dog barks*" if we replace "*then*", or become "*a dog barks* ***then*** *a man speaks*" if we invert the propositions between "*then*". We detailed the different words tested in Table 4.

Table 4: Detailed words used for Replace. BAT stands for Before-After-Test, seq2sup for sequence-to-superposition and sup2seq for superposition-to-sequence.

Set	Words	Replaced by one of
BAT	before after	after before
seq2sup	followed by, and then, then, before, after	as, while
sup2seq	as, while	followed by, and then, then, before, after

The Table 5 shows that our model perform very well at discriminate sound events relations, with 76.8% for the BAT, higher than the best of the compared study (68.5%). We can also see that our model perform very well on other tests which perturb the relations, with 90.6% when we invert the proposition after and before the sequence of words. It could imply that our model effectively captures the sequence and superposition relations. We also noticed for the Invert test with superposition words that our model is still able to detect the correct caption, probably because the first sound described in those sentences are the loudest or longest ones in the audio. Nevertheless, an AAC system requires computing the whole decoder pass-forward for each pair audio/caption, while usually ATR systems compute separate embeddings for each modality. For the A2T task, the post-processing is required to achieve an acceptable performance, requiring to keep the minimal and maximal value of the loss for each caption, or an estimation of them. If a new caption is added to the database, the minimal and maximal value also need to be computed or estimated with several audio files. This scaling should also be required for zero shot experiments, which is close to the A2T task.

Table 5: Accuracy over different perturbations on Clotho development-testing subset. 0.5 is the score of a random model.

System	Type	Set	Accuracy
MLP [28]	Replace	BAT	.496
MLP+ACBA [28]			.554
TFMER [28]			.509
TFMER+ACBA [28]			.685
CNext-trans (ours)	Replace	BAT	.768
		seq2sup	.825
		sup2seq	.903
CNext-trans (ours)	Invert	BAT	.892
		seq	.906
		sup	.778

6. CONCLUSIONS

In this study, we propose a straightforward method for leveraging any standard AAC system for A2T. We demonstrate that despite not being specifically trained for it, an AAC system can achieve reasonable performance on both the T2A and A2T subtasks. Furthermore, it can even attain state-of-the-art scores compared to ATR methods that do not employ external data. We also observed that our model often overestimates the loss value for a subset of captions in the A2T task, resulting in poor results in the initial configuration. To address this issue, we introduced a post-processing strategy based on min-max scaling to mitigate bias in the scores. This adjustment significantly improved the results, for instance, increasing R@1 from 0.038 to 0.148 on Clotho. Finally, we evaluated our system by perturbing the input captions and found that it outperforms another ATR method in distinguishing various sound event relations. In the future, potential research directions could involve modifying AAC training using a contrastive-based loss to enhance ATR performance or developing new benchmarks and test databases to refine the evaluation of ATR systems.

7. REFERENCES

[1] 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., July 2022.

[2] A. Krishnan, S. Rajesh, and S. SS, "Text-based image retrieval using captioning," in *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, 2021, pp. 1–5.

[3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 11 966–11 976.

[4] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[5] C. D. Kim, B. Kim, H. Lee, and G. 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132.

[6] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a ConvNeXt model to audio classification on AudioSet," 2023.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[8] D. Hendrycks and K. Gimpel, "Gaussian Error Linear Units (GELUs)," 2016.

[9] H. Zhang, M. Cissé, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[11] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*. ISCA, sep 2019.

[12] 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.

[13] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. L. Roux, and S. Watanabe, "BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up," DCASE2023 Challenge, Tech. Rep., May 2023.

[14] H. Won, B. Kim, I.-Y. Kwak, and C. Lim, "CAU submission to DCASE 2021 task6: Transformer followed by transfer learning for audio captioning," DCASE2021 Challenge, Tech. Rep., July 2021.

[15] E. Kim, J. Kim, Y. Oh, K. Kim, M. Park, J. Sim, J. Lee, and K. Lee, "Exploring train and test-time augmentations for audio-language learning," 2023.

[16] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an Audio Captioning Dataset," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020*. IEEE, 2020, pp. 736–740.

[17] M. Denkowski and A. Lavie, "Meteor Universal: Language Specific Translation Evaluation for Any Target Language," in *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014, pp. 376–380.

[18] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4566–4575.

[19] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic Propositional Image Caption Evaluation," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 382–398.

[20] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved Image Captioning via Policy Gradient optimization of SPIDEr," in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 873–881.

[21] 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.

[22] C.-C. Wang, J. Du, and J.-S. R. Jang, "Dcase 2023 task 6b: Text-to-audio retrieval using pretrained models," DCASE2023 Challenge, Tech. Rep., June 2023.

[23] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "On metric learning for audio-text cross-modal retrieval," 2022.

[24] Y. Xin, D. Yang, and Y. Zou, "Improving text-audio retrieval by text-aware attention pooling and prior matrix revised loss," 2023.

[25] P. Wang, S. Wang, J. Lin, S. Bai, X. Zhou, J. Zhou, X. Wang, and C. Zhou, "One-peace: Exploring one general representation model toward unlimited modalities," 2023.

[26] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, pp. 1–1, 2022.

[27] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, "Can audio captions be evaluated with image caption metrics?" in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 981–985.

[28] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salamon, "Audio-text models do not yet leverage natural language," 2023.