IRIT-UPS DCASE 2023 AUDIO CAPTIONING AND RETRIEVAL SYSTEM

Technical Report

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

[1]IRIT (UMR 5505), Université Paul Sabatier, CNRS, Toulouse, France
[2]Artificial and Natural Intelligence Toulouse Institute (ANITI)
{etienne.labbe,thomas.pellegrini,julien.pinquier}@irit.fr

ABSTRACT

This technical report provides a concise overview of our systems submitted to the DCASE Challenge 2023 for tasks 6a, ”Automated Audio Captioning” (AAC), and 6b, ”Language-Based Audio Retrieval” (LBAR). In task 6a, we made four distinct submissions. The first submission employed a standard CNN14 encoder paired with a transformer decoder. In the second submission, we replaced this encoder with a ConvNeXt model to enhance audio representation. The third submission incorporated additional training data. We introduced a new task embedding approach to differentiate between different writing styles and audio types. Finally, in the fourth submission, we employed an ensemble method to combine five models trained on different seeds, aiming to improve the quality of the captions. For task 6b, we use the AAC models and we propose a novel approach to accomplish the LBAR task by leveraging the AAC system loss function without requiring any additional training. Our most successful AAC and LBAR systems achieved a SPIDEr-FL score of 0.320 and an mAP@10 score of 0.269. These results demonstrate relative improvements of 22.6% and 21.2% compared to the AAC and LBAR baselines, respectively.

Index Terms— DCASE Challenge, audio captioning, text-to-audio retrieval, ConvNeXt, task embedding, ensemble learning

1. INTRODUCTION

The Automated Audio Captioning (AAC) and Language-Based Audio Retrieval (LBAR) tasks are multimodal tasks that use audio with natural language descriptions. AAC aims to build systems that described audio content, relations and attributes in a single sentence. On the other hand, the LBAR task is focused on retrieve a specific audio corresponding to a free-form description from a database of audios. The DCASE2023 challenge tasks 6a and 6b proposed to rank systems for these two audio-language tasks, and we propose to submit a single AAC model that can achieve both tasks.

For the AAC task, we employ a standard encoder-decoder architecture, with a pre-trained encoder for audio modelling and a transformer decoder to generate our captions. To improve our system, we added more data from other captioning datasets, add different data augmentations, improve beam search for inference and add a task embedding to our model to help caption generation when using different training datasets.

For the LBAR task, we proposed a novel strategy to rank audio files for each query by computing the AAC model loss to score each pair of audios and queries.

The source code will be available on GitHub [1] after the end of the challenge.

The rest of this paper is organized as follows: we start by describing our systems and experimental setup, then we present and comment the results in a second section, and we conclude in a last section.

2. SYSTEM DESCRIPTION

2.1. Architecture

To create a strong audio representation, we used the CNN14 pretrained architecture and weights [2] from Pretrained Audio Neural Network [1] (PANN) in our first submission. Then, we replace this encoder by the ConvNeXt [2] (CNext) model, a convolutional encoder originally created for image classification that we trained for audio tagging on AudioSet [3]. The details of the ConvNeXt training setup for audio tagging are given in [**?**]. In both cases, we precomputed the frame-level embeddings of shape for each audio file to speed up our training process. For an audio clip example of 10 seconds, we got a frame embedding of shape 768 × 31 with ConvNeXt and 2048 × 31 with CNN14.

The frame-level embeddings are given to a projection block, which maps the encoder features to the decoder part. More specifically, this block is formed by a sequence of dropout of 0.5, Linear layer, ReLU and dropout of 0.5.

The decoder part used is a standard transformer decoder architecture [4] with six decoder layers, four attention heads, a global embedding dimension set to 256, a feedforward dimension of 2048, a global dropout set to 0.2 and GELU [5] activation function.

2.2. Data augmentation

To improve model generalization and limit overfitting, we used three different augmentation methods:

- mixup [6] is applied to audio frame embeddings and input word embeddings with the hyperparameter α set to 0.4. Each embedding is mixed with another one of the current batch.
- label smoothing [7] modifies target labels by reducing the maximal probability to limit the confidence of the model.
- SpecAugment [8] is applied to audio frame level embeddings, with 6 stripes dropped of a maximal size of 4 for time dimen-

[1] `https://github.com/Labbeti/dcase2023challenge-task6ab`

[2] `https://zenodo.org/record/3987831`

sion and 2 stripes dropped with a maximal size of 2 for embedding dimension.

2.3. Datasets

In our experiments, we used one audio tagging dataset and four audio captioning datasets to train our systems.

AudioSet [3] (AS) is the largest publicly available audio tagging dataset, containing 2M pairs of 10-seconds audio clips and sound event classes. Like in almost all AAC systems, this dataset is used to pre-train the audio encoder to overcome the lack of audio captioning data.

Clotho [9] (CL) is an audio captioning dataset containing 6974 audio files between 15 and 30 seconds from the FreeSound website. Each audio is described by five different captions, written and corrected by humans to avoid grammatical errors and repetitions.

AudioCaps [10] (AC) is another audio captioning dataset containing 51308 audio files from AudioSet labeled with captions. Since original YouTube videos are removed, our version of the train split contains 46230 pairs of audio-captions.

Multi-Annotator Captioned Soundscapes [11] (MA) is the third audio captioning dataset containing 3930 files from the TAU Urban Acoustic Scene 2019 dataset. Each file last for 10 seconds and is described by at least 2 captions.

WavCaps [12] (WC) is a recent audio captioning dataset with 403050 pairs of audio-captions. The audio files are extracted from four different sources: AudioSet Strongly labeled subset, BBC Sound Effects, FreeSound and SoundBible websites, and the captions are post-processed by ChatGPT system.

The codebase used to download, read and extract data is named aac-datasets and available as a Pip package [3].

2.4. Data selection and pre-processing

In all our experiments, we excluded all the FreeSound data from WC since it contains a large overlap with CL (89%) and could also contain overlap with the private subset used to rank submitted systems in the DCASE challenge. We also filtered all the audios files which last for less than 0.5 seconds or more than 30 seconds.

The concatenation of the three additional audio captioning datasets (AC+MA+WC) results in 167203 new training files. However, not all added files are seen during an epoch. At the beginning of each epoch, we take the 3840 training files of CL and select randomly another 3840 files from the AC+MA+WC datasets, which results in 7680 audio training files per epoch.

Each audio file is resampled to 32kHz for MA and CL datasets. Captions are put in lowercase, and punctuation characters are removed. In addition, we fixed manually 996 invalid captions with grammatical and typographic errors in the AudioCaps training subset to improve train caption quality. When several references are available for a single audio file, we select randomly one of them at each epoch.

2.5. AAC Metrics

We used the five metrics required for the challenge to evaluate our systems. In the following section, we named “candidate” the caption predicted by an automatic system and “reference” the ground truth caption.

METEOR [13] computes the harmonic mean of precision and recall on the words of the candidate and reference. CIDEr-D [14] corresponds to the cosine similarity of the TF-IDF scores for the common n-grams in candidates and references. SPICE [15] is equal to the F-score of the vectorized representation of the semantic propositions extracted from the sentences using a dependency parser and handcrafted grammar rules. SPIDEr [16] average the CIDEr-D and SPICE scores to take into account both of their advantages. This metric is widely used to rank AAC systems.

We also used the FENSE metric [17], which computes the cosine similarity between the sentence embedding produced by a Sentence-BERT (SBERT) model combined with a fluency error detector which penalize captions that contain grammatical, syntactical errors or repetitions. The FENSE score of a sentence is equal to SBERT similarity when no error is detected, but it is divided by ten otherwise.

SPIDEr-FL [4] is a modification of SPIDEr proposed for the DCASE2023 challenge and combines SPIDEr score with the fluency error detector used in FENSE. This metric has been added to penalize models that used reinforcement learning on CIDEr-D, which usually leads to higher SPIDEr score but creates repetitive n-grams in candidates [18].

The codebase used to compute all of these metrics is named aac-metrics and available as a Pip package [5].

Table 1: Training and decoding hyperparameters.

Name	Value
Nb. Epochs (K)	400
Batch size	512
Optimizer	AdamW
Initial learning rate (lr_0)	$5 \cdot 10^{-4}$
β_1	0.9
β_2	0.999
ϵ	10^{-8}
Weight decay	2
Gradient clip norm value	1
Gradient clip norm type	L_2
mixup param. (α)	0.4
Label smoothing	0.2
Min prediction size	3
Max prediction size	40
Beam size	3

2.6. Task embedding

When we combined the captioning datasets listed in Section 2.3 for training our models, we found that the performance was not always better, and can even slightly decrease in some setups. This observation has already been seen before [20] and seems to indicate that the audio and caption domains of these datasets are different. As an example, the caption of CL are usually more diverse and complex than in AC.

To overcome this issue, we propose to add a task embedding to our models which encodes the dataset sources. More specifically,

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

[4] https://dcase.community/challenge2023/task-automated-audio-captioning#evaluation

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

N°	System	Training data	METEOR	CIDEr-D	SPICE	SPIDEr	SPIDEr-FL
-	Cross-referencing	N/A	0.305	0.903	0.231	0.567	0.563
-	DCASE2023 Baseline	CL	0.177	0.420	0.119	0.270	0.261
-	DCASE2022 Top-1 [19]	CL	0.186	0.513	0.126	0.320	N/A
1	CNN14-trans	CL	0.179	0.414	0.126	0.270	0.269
2	CNext-trans	CL	0.190	0.474	0.136	0.305	0.303
3	CNext-trans	CL+AC+MA+WC	0.192	0.485	0.139	0.312	0.310
4	CNext-trans (ensemble)	CL+AC+MA+WC	**0.193**	**0.500**	**0.140**	**0.320**	**0.320**

Table 2: Audio captioning results (Task 6a) on the development-testing subset of Clotho v2.1. Higher score is better. Reported values are the best over 5 seeds, except for our ensemble system (4).

N°	System	Training data	mAP@10	R@1	R@5	R@10
-	DCASE2023 Baseline	CL	0.222	0.130	0.343	0.480
-	DCASE2022 Top-1 [19]	CL+AC+MA	0.299	0.188	0.447	0.587
1	CNN14-trans	CL	0.186	0.106	0.288	0.419
2	CNext-trans	CL	0.231	0.140	0.353	0.483
3	CNext-trans	CL+AC+MA+WC	0.257	0.160	0.384	0.512
4	CNext-trans (ensemble)	CL+AC+MA+WC	**0.269**	**0.169**	**0.399**	**0.523**

Table 3: Audio retrieval results (Task 6b) on the development-testing subset of Clotho v2.1. Higher score is better. Reported values are the best over 5 seeds, except for our ensemble system (4).

we used four new Begin-Of-Sentence tokens, each one corresponding to a dataset (CL, AC, MA, WC). At training time, the token of the audio/reference caption source is given to the decoder. During inference, only the one corresponding to CL is used. A gain was obtained with this technique, compared to a simple dataset concatenation. We believe these source tokens may help in recognizing the audio events (that may be specific to a dataset source), and also in producing sentences that follow the expected writing styles of the CL dataset. This remains to be further studied.

2.7. Inference

We used the beam search algorithm to generate predictions during inference. Our version of beam search computes prediction per batch, which is much faster to generate and evaluate predictions during validation at each epoch. We also added several constraints to the beam search algorithm to forbid the model to predict the same word twice in a sentence. Only the words given by the pre-defined list of stop words of NLTK (like “the”, “and”, ...) can be repeated. We also limit the minimal and maximal prediction size to avoid some cases of degenerated candidates.

2.8. Hyperparameters

We select the best checkpoint over epochs using the FENSE validation score on the best beam search prediction since we found that it is more stable than the loss, SPIDEr or CIDEr-D which are dependent on the n-grams predicted in the candidates [21].

We detailed the optimization and inference hyperparameters values in the table 1. The weight decay is not applied to the bias weights of the network.

The learning rate is decreased during training at the end of each epoch k using a cosine scheduler rule:

$$\text{lr}_k = \frac{1}{2}\big(1 + \cos(\frac{k\pi}{K})\big)\text{lr}_0 \tag{1}$$

Before training, we pre-compute the frame-level audio embeddings to drastically speed up the captioning training phase. A single experiment runs on three hours with CL only and on single V100 graphics card.

2.9. Using a captioning system for retrieval

To use an AAC system without any training for LBAR task, we simply compute the cross entropy loss for each pair of audio and query. We believe that an AAC system should be able to give a higher loss value for incorrect queries, despite not having been trained with a contrastive loss. The decision rule to retrieval the best audio file from an audio database A with a query q and an AAC system f is given by the equation 2.

$$\text{Decision}(q, A, f) = \text{argmin}_{a \in A}\text{CE}(f(a, q_{\text{prev}}), q_{\text{next}}) \tag{2}$$

This method does not require any additional training to the AAC model and could be tested with any traditional AAC system. However, it can be slower than other retrieval system since we need to run the decoder forward for each pair of audios and queries.

2.10. Ensemble methods

In order to fuse our system outputs over several seeds, we employ different ensemble methods for each task. For the AAC task, we use a modified version of beam search, where the next token relies on the average logits of five models. For the LBAR task, we simply compute the average of the losses given by each model to score each pair of audio and query.

3. RESULTS

The AAC results are shown in table 2 and the LBAR ones in Table 3. We show our best of five seeds for each four submitted systems score in each task, compared the baseline of this year and to the best system of the previous year.

For the AAC task, the results show that our pre-trained encoder is much better than CNN14 one and drastically improve the metrics scores from +0.034 absolute SPIDEr-FL points. Adding more training data (1̃60K from AC+MA+WC compared to 3K from CL) with the task embedding option only improve by +0.007 points. An ensemble method of five models can improve even more, this score by +0.010 points, and becomes equal to the best system submitted last year for SPIDEr. However, the cross-referencing scores shows that we can still improve our systems to generate real human-like caption. We also noticed that the SPIDEr-FL and SPIDEr scores are really close, which means the model does not produce a lot of fluency errors.

For the LBAR task, we can see that the simple approach of using the cross entropy loss already performs well with the CNN14-trans system, and the scores are improved with the help of CNext, the additional data and the retrieval ensemble method. It implies that our AAC systems is able to achieve audio retrieval despite not having trained for it.

4. CONCLUSIONS

In this technical report, we presented our systems submitted to the DCASE 2023 challenge, with the use of a new pre-trained encoder, a task embedding method to help generation and a new way to achieve the LBAR task with an AAC system and without additional training. Further experiments could be done to deeply study the relationship between AAC and LBAR tasks, and maybe exclusively training the AAC system to discriminate audios files. We could also study the task embedding option and how this affects the caption generation and writing styles over the different datasets.

5. ACKNOWLEDGMENT

This work was partially supported by the Agence Nationale de la Recherche LUDAU (Lightly-supervised and Unsupervised Discovery of Audio Units using Deep Learning) project (ANR-18-CE23-0005-01). This work was granted access to the HPC resources of IDRIS under the allocation 2022-AD011013739 made by GENCI.

6. REFERENCES

[1] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, and M. 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, 01 2020. [Online]. Available: https://www.researchgate.net/publication/347217099_PANNs_Large-Scale_Pretrained_Audio_Neural_Networks_for_Audio_Pattern_Recognition

[2] 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. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2022/papers/Liu_A_ConvNet_for_the_2020s_CVPR_2022_paper.pdf

[3] 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. [Online]. Available: https://ieeexplore.ieee.org/document/7952261

[4] 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. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[5] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2016. [Online]. Available: https://arxiv.org/abs/1606.08415

[6] 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. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[7] 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. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Szegedy_Rethinking_the_Inception_CVPR_2016_paper.pdf

[8] 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. [Online]. Available: https://doi.org/10.21437%2Finterspeech.2019-2680

[9] 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. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9052990

[10] 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. [Online]. Available: https://aclanthology.org/N19-1011

[11] I. Martin and A. Mesaros, "Diversity and bias in audio captioning datasets," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 90–94. [Online]. Available: https://dcase.community/documents/workshop2021/proceedings/DCASE2021Workshop_Martin_34.pdf

[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. [Online]. Available: https://arxiv.org/pdf/2303.17395.pdf

[13] 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. [Online]. Available: http://aclweb.org/anthology/W14-3348

[14] 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. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Vedantam_CIDEr_Consensus-Based_Image_2015_CVPR_paper.pdf

[15] 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. [Online]. Available: https://panderson.me/images/SPICE.pdf

[16] 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. [Online]. Available: https://doi.org/10.1109/ICCV.2017.100

[17] 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. [Online]. Available: https://ieeexplore.ieee.org/document/9746427

[18] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. L. Tang, X. Shao, M. D. Plumbley, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," 2021. [Online]. Available: https://arxiv.org/abs/2108.02752

[19] 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. [Online]. Available: https://dcase.community/documents/challenge2022/technical_reports/DCASE2022_Xu_106_t6a.pdf

[20] J. Berg and K. Drossos, "Continual learning for automated audio captioning using the learning without forgetting approach," in *Proceedings of the 6th Detection and Classification of Acoustic Scenes and Events 2021 Workshop (DCASE2021)*, Barcelona, Spain, November 2021, pp. 140–144. [Online]. Available: https://dcase.community/documents/workshop2021/proceedings/DCASE2021Workshop_Berg_51.pdf

[21] E. Labbé, T. Pellegrini, and J. Pinquier, "Is my automatic audio captioning system so bad? spider-max: A metric to consider several caption candidates," in *Proceedings of the 7th Detection and Classification of Acoustic Scenes and Events 2022 Workshop (DCASE2022)*, Nancy, France, November 2022. [Online]. Available: https://dcase.community/documents/workshop2022/proceedings/DCASE2022Workshop_Labbe_46.pdf