EXPANDING ON ENCLAP WITH AUXILIARY RETRIEVAL MODEL FOR AUTOMATED AUDIO CAPTIONING

Technical Report

Jaeyeon Kim[1,2]*, Jaeyoon Jung*[2,3]*, Minjeong Jeon*[2]*, Sang Hoon Woo*[4]*, Jinjoo Lee*[2]*,*

[1] Seoul National Unversity, Seoul, Republic of Korea
[2] MAUM AI Inc., Seongnam, Republic of Korea,
[3] Soongsil University, Seoul, Republic of Korea,
[4] Independent Researcher
jaeyeonkim99@snu.ac.kr {jyjung, mjjeon, jjl}@maum.ai tonyswoo@gmail.com

ABSTRACT

In this technical report, we describe our submission to DCASE2024 Challenge Task6 (Automated Audio Captioning) and Task8 (Language-based Audio Retrieval). We develop our approach building upon the EnCLAP audio captioning framework and optimizing it for Task6 of the challenge. Notably, we outline the changes in the underlying components and the incorporation of the reranking process. Additionally, we submit a supplementary retriever model, a byproduct of our modified framework, to Task8. Our proposed systems achieve FENSE score of 0.542 on Task6 and mAP@10 score of 0.386 on Task8, significantly outperforming the baseline models.

Index Terms— Automated audio captioning, language-based audio retrieval, neural audio codec, audio-text joint embedding

1. INTRODUCTION

Automated audio captioning (AAC) refers to the cross-modal translation task of transcribing audio signals that contain sound events into concise and meaningful natural language descriptions [1]. Despite the recent success of deep learning in many traditional tasks, AAC remains a particularly challenging task, with substantial performance discrepancy between human and machine.

One significant contributor of the performance gap can be attributed to the intrinsic complexity of the task. Distinguishing between various sound events, especially between similar and ambiguous ones, requires extensive real-world knowledge. To mitigate this challenge, prior studies have incorporated additional real-world acoustic knowledge by employing pretrained audio encoders trained on audio classification tasks [2, 3, 4].

The scarcity of high-quality data poses an additional challenge in audio captioning. Notably, AudioCaps [5] and Clotho [6], two most widely used datasets for audio captioning, contain approximately 50K and 20K captions, respectively, while COCO captions [7], a widely used dataset for image captioning, has over 414K captions in its training set. Although Mei *et al.* [8] proposed WavCaps, a large-scale audio captioning dataset comparable in scale to COCO captions, it is important to note that WavCaps is a weakly-labeled dataset and cannot be considered a direct substitute for a high-quality dataset. To address this issue, previous works [9, 10, 11] have leveraged the text generation capabilities of pretrained language models like GPT-2 [12] and BART [13] to improve the semantic quality of the captions under data-scare scenarios. Additionally, some studies have also incorporated auxiliary loss terms, including keyword prediction loss [14] or sentence embedding loss [15], to provide additional training signal and improve the training procedure.

Expanding upon previous line of works, Kim *et al.* [16] proposed the EnCLAP framework, which integrates a set of pretrained models with an auxiliary training task. Notably, EnCLAP utilizes two acoustic feature encoders, EnCodec [17] and CLAP [18], to generate timestep-level and sequence-level representation, of the input audio sequence, respectively. For caption decoder, the framework employs a pretrained BART model [13]. Furthermore, Kim *et al.* also introduced masked codec modeling (MCM), an auxiliary task designed to enhance the acoustic awareness of the caption decoder. The combination of these approaches allowed EnCLAP to achieve state-of-the-art performance on the AudioCaps dataset.

In this work, we adapt the EnCLAP framework to tackle DCASE2024 Challenge. We aim to optimize and enhance the performance of each component within the EnCLAP framework, while adhering to the challenge's rules and regulations. Specifically, we investigate alternative models for the EnCodec and CLAP components and adopt a sampling and reranking procedure to further improve the quality of the generated captions. We submit our resulting system to Task6 and Task8 of the challenge.

2. METHOD

2.1. Neural Audio Codec

Neural audio codecs are autoencoder models designed to encode waveforms into sequences of discrete codes. Recent advancements [19, 17, 20] typically employ residual vector quantization (RVQ) for compression, utilizing multiple codebooks to quantize the residuals of preceding codebooks. Ultimately, the input waveforms are transformed into a set of parallel discrete code sequences, each of which is associated with a unique codebook. Neural audio codecs have demonstrated success as the acoustic representation format in generative audio models [21, 22, 23].

In the context of audio captioning, EnCLAP [16] employs the neural audio codec, specifically EnCodec [17], to represent the input waveform at the timestep level. This approach is based on the assumption that pretrained language models are better suited to process discrete inputs compared to continuous ones. In this work, we replace EnCodec in the original EnCLAP framework with De-

Table 1: Results on Task8 Language-Based Audio Retrieval on Clotho evaluation split. CL, AC, and WC refer to Clotho, AudioCaps, and WavCaps, respectively.

Model	Audio Encoder	Text Encoder	mAP@10	R@1	R@5	R@10
Baseline	CNN14	all-mpnet-base-v2	0.222	0.130	0.343	0.480
Pretrained on CL + AC + WC						
Pretrain 1	CNext	bge-base	0.334	0.222	0.485	0.619
Pretrain 2	CNext	bert-base	0.325	0.208	0.479	0.618
Pretrain 3	CNext	roberta-base	0.326	0.214	0.474	0.614
Pretrain 4	CNext	bge-large	0.339	0.219	**0.502**	0.633
Pretrain 5	CNext	bert-large	0.339	0.220	0.500	**0.635**
Pretrain 6	CNext	roberta-large	**0.342**	**0.228**	0.492	0.632
Finetuned on CL						
Finetune 1	CNext	bge-base	0.356	0.235	0.522	0.649
Finetune 2	CNext	bert-base	0.356	0.235	0.520	0.654
Finetune 3	CNext	roberta-base	0.365	0.250	0.523	0.659
Finetune 4	CNext	bge-large	0.369	0.249	0.530	0.665
Finetune 5	CNext	bert-large	0.367	0.247	0.526	0.663
Finetune 6	CNext	roberta-large	0.375	0.256	0.535	0.669
Ensemble 1			0.385	0.265	**0.547**	0.676
Ensemble 2			**0.386**	**0.267**	**0.547**	**0.680**
Ensemble 3			0.378	0.257	0.543	0.676

script Audio Codec (DAC) [20], as DAC has demonstrated superior performance in audio compression, as well as downstream tasks [20, 24].

2.2. Audio-Text Joint Embedding

The original EnCLAP employs CLAP [18] embeddings as the sequence-level acoustic representation of the input audio. However, due to potential overlap between the training dataset of CLAP and the evaluation dataset, we substitute CLAP with an alternative model. Specifically, we utilize the audio encoder of the baseline model of the challenge, hereinafter referred to as CNext [25], which was trained on the AudioSet [26] dataset for the audio classification task. In our preliminary experiments, we observed that the variant of CNext finetuned using the audio-text retrieval task exhibits superior performance. Therefore, we adopt this variant in our work.

2.3. Generation and Reranking

Previous works, including EnCLAP [16], have utilized beam search decoding for caption generation. However, Wu *et al.* [4] showed that the sampling-then-reranking approach yields more diverse and informative captions. Therefore, we adopt the approach proposed by Wu *et al.* [4], where we generate a set of candidate captions through nucleus sampling and select the most suitable one via reranking.

Our candidate selection procedure is a two-stage process. First, we use the FENSE fluency error detector [27] to filter captions containing fluency errors. We then rank the remaining candidates based on the weighted sum of two reranking scores: encoder reranking and decoder reranking. The encoder reranking score is cosine similarity score between the input audio representation and the generated caption representation computed using the retriever model described in Section 2.2. For the decoder reranking score, we use the log-likelihood of the generated caption given the input audio.

3. EXPERIMENT

We assess the performance of our modified EnCLAP model on Task6 of DCASE2024 Challenge and report the results. Additionally, we evaluate the retriever model described in Section 2.2 on Task8 of the challenge.

3.1. Setup

Dataset. In our experiment, we adopt a two-stage training process, where we pretrain on a larger dataset and subsequently finetune on a smaller dataset. The pretraining dataset comprises a combination of AudioCaps [5], WavCaps [8], Clotho [6], and Clotho-ChatGPT-Mixup [4]. Conversely, the finetuning dataset consists solely of the Clotho dataset. To comply with the challenge regulations, we exclude any potential overlapping data from Freesound in the WavCaps dataset. Additionally, we only use audio clips with durations between 1 and 30 seconds from the WavCaps dataset. For Clotho, we utilize only the training split of the dataset.

Model Configuration. From the original EnCLAP model configuration, we experiment only with the EnCLAP-large setup to maximize the performance of the final model. We use a variant of the DAC model that transforms a 24kHz acoustic waveform to 75Hz code sequences, with 32 codes per timeframe and codebook size of 1024. For the retriever model, we initialize the audio encoder with CNext-tiny by Pelligrini *et al.* [25]. For the text encoder, we experiment with 6 different pretrained language models: BGE-base [28], BERT-base [29], RoBERTa-base [30], BGE-large [28], BERT-large [29], RoBERTa-large [30]. For the sequence-level feature encoder, we choose Pretrain 1 version listed in Table 1. Note that we resample the input audio to appropriate sample rates before processing it with feature encoders.

Generation. We use nucleus sampling with a probability threshold of 0.95 and a temperature of 0.5 to generate 30 candidates. We rank the candidates by the weighted sum of the encoder reranking score and the decoder reranking score using weights of 0.7 and 0.3, respectively.

Table 2: Results on Task6 Automated Audio Captioning on Clotho evaluation split. For EnCLAP-large, we report the scores using the official *clotho-finetune-large* checkpoint, which was pretrained on the AudioCaps dataset and finetuned on the Clotho dataset.

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL	Vocabulary	FENSE
Baseline	0.1897	0.4619	0.1335	0.2977	0.2962	551	0.5040
EnCLAP-large	0.1864	0.4641	0.1336	0.2989	0.2971	592	0.5116
Submission1	0.1989	**0.4826**	0.1483	**0.3155**	**0.3155**	840	0.5386
Submission2	0.1955	0.4775	0.1423	0.3099	0.3099	**865**	0.5419
Submission3	**0.2003**	0.4780	**0.1488**	0.3134	0.3134	825	0.5393
Submission4	0.1994	0.4778	**0.1488**	0.3133	0.3133	815	**0.5420**

3.2. Training

Language-Based Audio Retrieval. We train our retriever models using the m-LTM framework [31], a learning-to-match framework for the minibatch setting, designed to minimize the modality gap between audio and text embedding in audio-text retrieval tasks. During the pretraining phase, we use a mixed dataset comprising AudioCaps [5], WavCaps [8], and Clotho [6].

Automated Audio Captioning. For audio caption training, we follow the original EnCLAP setup and use a combination of two tasks: captioning task and MCM task. MCM is an auxiliary training task which involves masking a part of the input codec sequence and predicting it, analogous to the masked language modeling (MLM) approach. Note that we omit the MCM task during the finetuning stage. During the pretraining stage, we use a combined dataset of AudioCaps [5], WavCaps [8], and Clotho-ChatGPT-Mixup [4].

3.3. Results

Language-based Audio Retrieval. We present the results of our evaluation on Task8 in Table 1. Our models significantly outperform the baseline. We also find that ensembling models with different encoders yields additional score improvements. For the challenge, we submit the following four models:

1. **Finetune 1:** CNext audio encoder and RoBERTa-large text encoder
2. **Ensemble 1:** An ensemble of the top 3 fine-tuned models: Finetune 4, 5, 6
3. **Ensemble 2:** An ensemble of all fine-tuned models
4. **Ensemble 3:** An ensemble of all pre-trained and fine-tuned models

Automated Audio Captioning. We summarize the results of our evaluation on Task6 in Table 2. Our models surpass both the DCASE2024 baseline and EnCLAP-large by a wide margin. The details of our submissions are as follows:

1. **Submission 1:** A modified EnCLAP model with DAC and CNext audio-text joint embedding
2. **Submission 2:** An average soup [32] model of 5 modified EnCLAP models
3. **Submission 3:** An ensemble of 7 modified EnCLAP models
4. **Submission 4:** An ensemble of 7 modified EnCLAP models and 2 soup models

4. CONCLUSION

This report outlines our approach to DCASE2024 Challenge. We investigate the application of the recently introduced m-LTM loss to language-based text retrieval. For automated audio captioning, we attempt to optimize and improve the the EnCLAP framework by introducing new backbone models, that is, specifically by replacing EnCodec with DAC and CLAP with the audio encoder from the aforementioned retriever. We also integrate a sampling-and-reranking scheme to the generation procedure. In our future work, we hope to investigate the reciprocal effects of captioning and retrieval tasks.

5. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017.

[2] X. M. et al., "Audio captioning transformer," in *DCASE Workshop*, 2021.

[3] E. Labbé, T. Pellegrini, and J. Pinquier, "Conette: An efficient audio captioning system leveraging multiple datasets with task embedding," *arXiv preprint arXiv:2309.00454*, 2023.

[4] S.-L. Wu, X. Chang, G. Wichern, J.-W. Jung, F. Germain, J. Le Roux, and S. Watanabe, "Improving audio captioning models with fine-grained audio features, text embedding supervision, and llm mix-up augmentation," in *ICASSP*, 2024.

[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *NAACL*, 2019.

[6] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *ICASSP*, 2020.

[7] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick, "Microsoft coco captions: Data collection and evaluation server," *arXiv:1504.00325*, 2015.

[8] 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:2303.17395*, 2023.

[9] M. Kim, S.-B. Kim, and T.-H. Oh, "Prefix tuning for automated audio captioning," in *ICASSP*, 2023.

[10] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang, "Pengi: An audio language model for audio tasks," *arXiv:2305.11834*, 2023.

[11] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *DCASE Workshop*, 2021.

[12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, 2019.

[13] 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 *ACL*, 2020.

[14] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A Transformer-Based Audio Captioning Model with Keyword Estimation," in *INTERSPEECH*, 2020.

[15] E. Labbé, J. Pinquier, and T. Pellegrini, "Multitask learning in audio captioning: a sentence embedding regression loss acts as a regularizer," *arXiv preprint arXiv:2305.01482*, 2023.

[16] J. Kim, J. Jung, J. Lee, and S. H. Woo, "Enclap: Combining neural audio codec and audio-text joint embedding for automated audio captioning," in *ICASSP*, 2024.

[17] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv:2210.13438*, 2022.

[18] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP*, 2023.

[19] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM TASLP*, 2021.

[20] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," in *NeurIPS*, 2023.

[21] F. Kreuk, G. Synnaeve, A. Polyak, U. Singer, A. Défossez, J. Copet, D. Parikh, Y. Taigman, and Y. Adi, "Audiogen: Textually guided audio generation," in *ICLR*, 2022.

[22] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, *et al.*, "Neural codec language models are zero-shot text to speech synthesizers," *arXiv:2301.02111*, 2023.

[23] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Defossez, "Simple and controllable music generation," in *NeurIPS*, 2023.

[24] H. Wu, H.-L. Chung, Y.-C. Lin, Y.-K. Wu, X. Chen, Y.-C. Pai, H.-H. Wang, K.-W. Chang, A. H. Liu, and H.-Y. Lee, "Codec-superb: An in-depth analysis of sound codec models," *arXiv preprint arXiv:2402.13071*, 2024.

[25] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a convnext model to audio classification on audioset," *arXiv:2306.00830*, 2023.

[26] 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 *ICASSP*, 2017.

[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.

[28] S. Xiao, Z. Liu, P. Zhang, and N. Muennighof, "C-pack: Packaged resources to advance general chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.

[29] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019.

[30] 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," *arXiv preprint arXiv:1907.11692*, 2019.

[31] M. Luong, K. Nguyen, N. Ho, R. Haf, D. Phung, and L. Qu, "Revisiting deep audio-text retrieval through the lens of transportation," in *ICLR*, 2024.

[32] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt, "Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time," in *ICML*, 2022.