LEVERAGING CED ENCODER AND LARGE LANGUAGE MODELS FOR AUTOMATED AUDIO CAPTIONING

Technical Report

Jizhong Liu, Gang Li, Junbo Zhang, Chenyu Liu, Heinrich Dinkel†, Yongqing Wang†, Zhiyong Yan†, Yujun Wang, Bin Wang*

AI Lab, Xiaomi Corporation, China
{liujizhong1, ligang5, zhangjunbo1, liuchenyu3}@xiaomi.com

ABSTRACT

This technical report presents an automated audio captioning (AAC) method participating in the DCASE 2024 Challenge Task 6. The method builds upon our previous work [1][1]. Recent advancements in large language models (LLMs), coupled with improved training approaches for audio encoders, have opened up possibilities for enhancing AAC. Thus, we optimize AAC from three points: 1) a pre-trained audio encoder named consistent ensemble distillation (CED) improves the effectivity of acoustic tokens, with a querying transformer (Q-Former) bridging the modality gap to LLM and compress acoustic tokens; 2) we introduce a Llama 2 with 7B parameters as the decoder; 3) a frozen Llama 3 Instruct with 8B parameters corrects text errors caused by insufficient training data and annotation ambiguities. Both the encoder and text decoder are optimized by low-rank adaptation (LoRA). Our method obtains a 53.2 FENSE score.

***Index Terms*—** AAC, CED, LLM, LoRA, Q-Former, acoustic token, error correction

1. INTRODUCTION

Automated audio captioning (AAC) is a multimodal task to describe the audio content in natural language [2]. Distinct from speech-to-text conversion, the AAC system implements the audio-to-text conversion to capture the underlying acoustic semantic information. AAC studies have gathered increasing interest in recent years, driven by the rising demand for intelligent interactions and information retrievals.

In recent studies, the typical encoder-decoder architecture has been progressively constructed [2]. The audio encoder extracts acoustic tokens from the input audio, while the text decoder generates the caption based on acoustic tokens. Generally, audio or speech extractors serve as the encoder (e.g., PANNs [3], BEATs [4], SpeechT5 [5], and Whisper [6]), and language models serve as the decoder (e.g., BERT [7], GPT-2 [8], and BART [9]). Despite various encoder-decoder combinations, state-of-the-arts (SOTAs) consistently leverage pre-trained models. For instance, the winner of the DCASE 2023 Challenge [10] uses a BEATs-BART architecture.

In this technical report, we present our method sharing a similar architecture with current mainstream methods. The architecture combines an audio encoder and a text decoder. Innovatively, a post-corrector is used to corrects text errors caused by insufficient training data and annotation ambiguities.

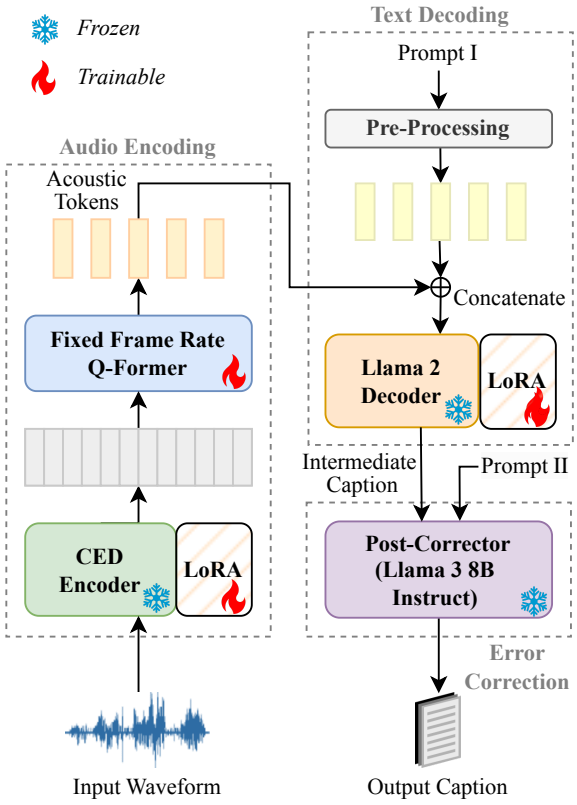

Figure 1: Architecture of the proposed method.

The information on the core components is as follows:

- **Audio encoder:** consistent ensemble distillation (CED) [11] with querying transformer (Q-Former) [12].
- **Text decoder:** Llama 2 with 7B parameters [13].
- **Post-corrector:** Llama 3 Instruct with 8B parameters [14].

This method is based on our previous work [1], wherein experiments shows that each component is effective and collective. Our method obtains a 53.2 FENSE score on Clotho evaluation split.

*Corresponding author.
†Equal contribution.
[1]Available: https://github.com/frankenliu/LOAE

2. METHODOLOGY

As illustrated in Figure 1, the proposed method is an encoder-decoder architecture.

2.1. Audio Encoding

CED is an Audio tagging model with higher mean average precision (mAP), lower computational complexity, and fewer output tokens than the widely used Bidirectional Encoder representation from Audio Transformers (BEATs) [4]. It employs a simple training framework on distilling student models from large teacher ensembles with consistent teaching [11]. A pre-trained CED model (without the output layer) serve as the audio encoder and is fine-tuned using low-rank adaptation (LoRA), a parameter-efficient fine-tuning approach for Transformer [15]. When processing the same audio clip, CED generates only approximately half the number of tokens compared to other models (e.g., BEATs and Whisper). Producing 248 tokens per 10 seconds with CED still amounts to a relatively large data for decoding. To bridge the modality gap, every 17 tokens are compressed to 1 token using Q-Former, which enhances encoding attention and decreases decoding complexity. The final number of acoustic tokens is equal to those processed by the 14-layer convolutional neural network (CNN14) [3].

2.2. Text Decoding

A regular language tokenizer performs the pre-processing operation. Subsequently, the Llama 2 decoder with 7B parameters is also fine-tuned using LoRA to tailor it the downstream task. This refined decoder is capable of producing more precise captions by leveraging optimized audio encoding and the deeper understanding provided by LLMs. As described in Table 1 (`Prompt I` in Figure 1), an instruction prompt guides Llama 2 in understanding AAC tasks. `<AcousticTokens>` denotes acoustic tokens, which are text-like tokens directly embedded into text tokens.

2.3. Error Correction

Under current conditions of insufficient training data and annotation ambiguities, the text decoder may learn incorrect patterns, such as single phrase loops and grammatical errors. While data augmentation can mitigate errors to some extent, linguistic errors may still persist. Thus, a frozen Llama 3 Instruct with 8B parameters is employed during the inference stage to correct linguistic errors. In this work, the post-corrector is activated only when the error probability exceeds the threshold (90%) set by the Error Detector [16]. To guide the error correction process, the instruction prompt (`Prompt II`) contains a sentence that does not conform to linguistic conventions, and `<Text>` denotes the input sentence (intermediate caption in Figure 1) of the post-corrector.

2.4. Loss Function

The training of the our network is divided into the pre-training and fine-tuning stages. In both stages, cross-entropy is used as the loss function.

$$\mathcal{L}_{CE} = -\frac{1}{N}\sum_{n=1}^{N}\log p\left(y_n \mid y_{1:n-1}, \mathbf{X}\right) \quad (1)$$

where $\mathbf{X}$, N and y_n denotes input audio, the number of ground truth tokens, and n-th token in a caption, respectively.

Table 1: Instruction prompts of the proposed method.

Prompt I	Describe the detail of this audio: <AcousticTokens> \n — \n Detailed:
Prompt II	System: You are a language specialist who can revise the sentence to make it more correct and idiomatic! You should follow the below format: rain is falling on a tin roof ==> rain is falling on the tin roof User: <text>

2.5. Data Augmentation

Alongside Clotho and AudioCaps, the WavCaps dataset [17] is introduced for data augmentation. Since Clotho is extracted from Freesound website, the forbidden clips in WavCaps has been excluded. WavCaps is a large-scale weakly labeled audio captioning dataset comprising approximately 400k audio clips with paired captions sourced from AudioSet [18], BBC sound effects[2], FreeSound[3], and SoundBible[4]. The captions in WavCaps are based on a three-stage processing pipeline by ChatGPT [19]. The audio clips from all three datasets are uniformly cropped into 10 seconds for training purposes. During the fine-tuning stage, the model is separately trained on Clotho.

3. EXPERIMENTS

3.1. Evaluation Dataset

The evaluations are primarily based on the Clotho v2.1 dataset [20], recognized as the the most authoritative dataset and the benchmark for ranking in the DCASE 2024 Challenge. The audio clips are no longer than 30 seconds and the captions contains 8 to 20 words. The dataset is split into the development, validation, evaluation, and testing subsets. Performance comparisons are conducted on Clotho evaluation split.

3.2. Metrics

The experiments refer to almost all metrics in DCASE 2024 Task 6, including METEOR [21], CIDEr [22], SPICE [23], SPIDEr [24], SPIDEr-FL [16], Sentence-BERT [25], and FENSE [16]. The ranking metric is FENSE. METEOR and CIDEr are both based on n-gram overlap, while SPICE focuses on the overlap computed on semantic graphs constructed by objects, attributes, and relations. SPIDEr is the mean of CIDEr and SPICE. SPIDEr-FL utilizes the Error Detector of FENSE to penalize the SPIDEr score of the sentence with an error probability greater than 90%. Sentence-BERT and FENSE are both BERT-based models to evaluate the similarity between the ground truth and the generated caption, with FENSE incorporating an Error Detector to penalize erroneous sentences. Additionally, BLEU [26] and ROUGE-L [27], two supplementary metrics, are included in ablation studies to enhance the credibility. In Tables 2, B1 to B4, RG, ME, CD, SP, SD, SD-F, SB, and FS denote BLEU-1 to BLEU-4, ROUGE-L, METEOR, CIDEr, SPICE, SPIDEr, SPIDEr-FL, Sentence-BERT, and FENSE. All scores are multiplied by 100.

[2]https://sound-effects.bbcrewind.co.uk/

[3]https://freesound.org/

[4]https://soundbible.com/

Table 2: Performance Comparison on Clotho evaluation split.

Sysytem ID	Encoder	Decoder	B1	B2	B3	B4	RG	ME	CD	SP	SD	SD-F	SB	FS
Baseline	ConvNeXt	Transformer	59.5	39.2	26.0	17.0	39.3	19.0	46.2	13.4	29.8	29.6	50.6	50.4
Submission 1	Dasheng-Base	Llama 2-7B	59.9	39.6	26.7	17.6	44.4	19.1	49.9	13.9	31.9	31.8	52.3	52.2
Submission 2	Dasheng-0.6B	Llama 2-7B	59.3	38.8	25.4	16.3	43.4	18.7	45.8	13.7	29.8	29.7	52.1	52.0
Submission 3	CED-Base	BART-Base	57.1	38.1	25.4	16.3	44.3	17.8	44.7	12.7	28.7	28.7	52.1	52.1
Submission 4	CED-Base	Llama 2-7B	**60.6**	**40.4**	**27.2**	**17.8**	**44.5**	**19.4**	**50.3**	**14.5**	**32.4**	**32.3**	**53.4**	**53.2**

3.3. Implementation Details

We train our models on two NVIDIA A100 (80 GB) GPUs with the Trainer accelerator[5]. The models are optimized on an AdamW optimizer [28] with a weight decay coefficient of 1×10^{-6} and warming up first 0.3 epochs. We pre-train the models with 10 epochs, a batch size of 48, and a learning rate of 5×10^{-5}, while we fine-tune the models with 20 epochs, a batch size of 10, and a learning rate of 5×10^{-6}. LoRA matrices are added to the "q" and "v" of the Transformer architecture. The audio sampling rate is 160000 *Hz*.

3.4. Results

The experiments include three different encoders: CED-Base [11][6], Dasheng-Base [29][7],and Dasheng-0.6B [29][7]. It also include two different decoders, including Llama 2-7B [13] and BART-Base [9]. The scores of baseline [30, 31] is obtained from the official website[8].

The experimental results is shown in Table 2. LoRA and QFormer are used in all experiments, so the relevant information is omitted in the table. The details of the our AAC systems are as follows:

- **Submission 1:** A Dasheng-Llama architecture, which is a single model with Dasheng-Base and Llama 2-7B.
- **Submission 2:** A Dasheng-Llama architecture, which is a single model with Dasheng is the middle version and Llama 2-7B.
- **Submission 3:** A CED-BART architecture,, which is a single model with CED-Base and BART-Base.
- **Submission 4:** A CED-Llama architecture, which is a single model with CED-Base and Llama 2-7B.

Therefore, the best version is Submission 4 with a 53.2 FENSE score.

4. CONCLUSION

This technical report describes several AAC systems for DCASE 2024 Task 6. The best version is Submission 4. This method builds upon our previous work [1]. To optimize audio encoding, we combine CED-based encoder with LoRA. To optimize text decoding, we also fine-tune a Llama 2-7B with LoRA. Q-Former connects the audio encoder and the text decoder, building an effective bridge to capture and represent the underlying acoustic features while reducing decoding complexity. A frozen Llama 3-8B-Instruct corrects text errors caused by insufficient training data and annotation ambiguities. The proposed method obtains a 53.2 FENSE score on Clotho evaluation split.

[5]https://huggingface.co/docs/transformers/trainer
[6]https://github.com/RicherMans/CED
[7]https://github.com/RicherMans/Dasheng
[8]https://github.com/Labbeti/dcase2024-task6-baseline

5. REFERENCES

[1] J. Liu, G. Li, j. Zhang, H. Dinkel, Y. Wang, Z. Yan, Y. Wang, and B. Wang, "Enhancing automated audio captioning via large language models with optimized audio encoding," in *Proc. Interspeech*, 2024, (Accpeted).

[2] X. Xu, Z. Xie, M. Wu, and K. Yu, "Beyond the status quo: A contemporary survey of advances and challenges in audio captioning," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 95–112, 2024.

[3] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[4] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," *arXiv preprint arXiv:2212.09058*, 2022.

[5] J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang, *et al.*, "SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing," *arXiv preprint arXiv:2110.07205*, 2021.

[6] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. Mcleavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning (ICML)*, vol. 202, 23-29 Jul 2023, pp. 28 492–28 518.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[9] 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," *arXiv preprint arXiv:1910.13461*, 2019.

[10] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. L. Roux, and S. Watanabe, "Improving audio captioning models with fine-grained audio features, text embedding supervision, and LLM mix-up augmentation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 316–320.

[11] H. Dinkel, Y. Wang, Z. Yan, J. Zhang, and Y. Wang, "CED: Consistent ensemble distillation for audio tagging," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 291–295.

[12] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[13] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "LLAMA 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[14] AI@Meta, "Llama 3 model card," 2024. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

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

[17] 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.

[18] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[19] "Introducing ChatGPT," OpenAI Blog, 2022. [Online]. Available: https://openai.com/blog/chatgpt

[20] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 736–740.

[21] A. Lavie and A. Agarwal, "METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments," in *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007, p. 228–231.

[22] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.

[23] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *European Conference on Computer Vision (ECCV)*, 2016, pp. 382–398.

[24] 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, pp. 873–881.

[25] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Nov 2019, pp. 3982–3992.

[26] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceeds of the 40th Annual Meeting on Association for Computational Linguistics*, 2002, p. 311–318.

[27] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, Jul 2004, pp. 74–81.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[29] H. Dinkel, Z. Yan, Y. Wang, J. Zhang, Y. Wang, and B. Wang, "Scaling up masked audio encoder learning for general audio classification," in *Proc. Interspeech*, 2024, (Accpeted).

[30] 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.

[31] T. Pellegrini, I. Khalfaoui-Hassani, E. Labbé, and T. Masquelier, "Adapting a ConvNeXt Model to Audio Classification on AudioSet," in *Proc. Interspeech*, 2023, pp. 4169–4173.