SJTU-THU AUTOMATED AUDIO CAPTIONING SYSTEM FOR DCASE 2024

Technical Report

Wenxi Chen[1]*, Xiquan Li*[1]*, Ziyang Ma*[1]*, Yuzhe Liang*[1]*, Zhisheng Zheng*[1]*, Anbai Jiang*[2]
Yanmin Qian[1]*, Pingyi Fan*[2]*, Wei-Qiang Zhang*[2]*, Cheng Lu*[3]*, Jia Liu*[2]*, Xie Chen*[1]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2] Department of Electronic Engineering, Tsinghua University, Beijing, China
[3] School of Economics and Management, North China Electric Power University, Beijing, China
{1029713857, chenxie95}@sjtu.edu.cn

ABSTRACT

Task 6 (Automated Audio Captioning) of the DCASE 2024 Challenge requires the automatic creation of textual descriptions for general audio signals. This technical report presents a novel model that integrates a self-supervised model with a large language model (LLM) for audio captioning. For audio feature extraction, we utilize the efficient self-supervised pre-trained model, EAT, to achieve more effective audio representation extraction. The language model component is based on Vicuna, a large language model, which we fine-tune using LoRA to fully harness its robust reasoning capabilities. During training, linear layers function as projectors to align audio and textual representations. Our model is pre-trained using the Clotho, WavCaps, AudioCaps, and MACS datasets, and fine-tuned on Clotho. For decoding, we employ a filtering strategy based on the CLAP model. By leveraging the text-audio alignment capabilities of the CLAP model, we filter out the beam search decoding results to retain only the textual description that best matches the input audio. Evaluation on the testing subset of Clotho demonstrates that our model achieves a FENSE score of 0.5431 in the single-system setting and 0.5429 in the multi-system setting, while the multi-systems outperform the single-system in other metrics. Our project code is based on the SLAM-LLM toolkit [1].

***Index Terms*—** Audio captioning, EAT, Large language model, CLAP, Model ensembling

1. INTRODUCTION

Automated Audio Captioning (AAC) is a multimodal challenge that involves generating corresponding textual content descriptions from input audio data. In recent years, the Detection and Classification of Acoustic Scenes and Events (DCASE) challenge has significantly advanced the field of AAC. Many methods proposed in the competition [1, 2, 3] have provided valuable insights and innovations in AAC. Additionally, the release of audio-text paired datasets such as AudioCaps [4], Clotho [5], MACS [6], and WavCaps [7] has further propelled research in AAC.

Historically, AAC models have relied on language models like standard Transformer decoders [8] and BART decoders [9] to generate textual content descriptions. However, the captions generated by these models often perform poorly on text quality-related metrics such as SPIDEr-FL and FENSE [10], resulting in less fluent text. Recently, the advent of large language models (LLMs) such as GPT-2 [11], FlanT5 [12], and LLaMA [13] has led to significant advancements in various cross-modal understanding tasks, including image captioning [14] and video captioning [15]. These models have outperformed their smaller parameter predecessors. Motivated by these advancements, we introduced LLM into our system for audio captioning tasks. By leveraging the large language model Vicuna [16], our model aims to generate higher-quality text.

We employ the EAT model [17] as the audio encoder to extract audio features, leveraging its self-supervised learning capabilities known for high efficiency and effectiveness. Efficiently pre-trained on the Audioset dataset [18], EAT has demonstrated state-of-the-art performance in audio classification tasks such as AS-2M and AS-20K. After encoding the input audio's mel-spectrogram into audio representations using EAT, we utilize lightweight linear layers as projectors to align these representations with text embeddings. Subsequently, we use Vicuna to attend to those representations and generate corresponding text descriptions.

During training, to balance efficiency and cost, our model employs LoRA [19] adapters for parameter-efficient fine-tuning (PEFT) of the large language model (LLM) while freezing the EAT model and training only the linear layers used for alignment. In the decoding stage, we utilize the CLAP model to filter beam search results. By leveraging the CLAP model's capability to evaluate text-audio similarity, we calculate the similarity scores between multiple beam search decoded audio captions and the input audio. We then filter out the captions based on these similarity scores, ultimately retaining the highest-scoring caption as the final result.

2. SYSTEM DESCRIPTION

2.1. Network Architecture

Our model employs the EAT model [17], a Transformer-based architecture, as its audio encoder to extract audio features. The EAT model is an efficient self-supervised pre-training framework that utilizes masked language modeling as a pretext task within a bootstrap methodology [20]. Compared to other self-supervised learning models such as BEATs [21] and Audio-MAE [22], the EAT model achieves over ten times greater pre-training efficiency on the Audioset dataset [18]. Furthermore, the EAT model exhibits substantial performance enhancements over supervised models like PANNs [23] and AST [24], as well as other unsupervised models like BEATs and Audio-MAE, particularly in audio classification

[1] https://github.com/X-LANCE/SLAM-LLM

tasks including AS-2M, AS-20K, and ESC-50 [25].

In our experiments, we utilize the EAT-base model [2] which has been fine-tuned on the AS-2M dataset to extract audio representations. Specifically, the EAT model converts waveforms into Mel-spectrograms, transforms them into patch embeddings using a CNN encoder, and extracts audio representations, E_A, at approximately 50Hz via a standard 12-layer ViT-B [26] module.

Inspired by the previous system [1] in the DCASE challenge, we use lightweight linear layers to downsample the extracted audio representations and align them with the text embeddings. Specifically, we apply a 5x downsampling using two linear layers to convert the audio representation E_A into E'_A in our experiments.

Unlike the systems submitted in previous DCASE competitions, our approach employs the large language model Vicuna [3] [16] as the text decoder. Inspired by the application of large language models in understanding tasks across speech and audio modalities, such as SLAM-ASR [27] and BAT [28], we concatenate the embeddings of the text modality with the transformed audio representations to obtain the joint representations. Specifically, our model processes textual prompts and audio captions using Vicuna's default tokenizer, producing the corresponding text embeddings E_P and E_T. The joint representation E_J is formed by concatenating these embeddings as follows:

$$E_J = \begin{cases} [E'_A; E_P; E_T] & \text{during training} \\ [E'_A; E_P] & \text{during inference} \end{cases} \tag{1}$$

In our experiments, we employ a simple prompt, "Describe the audio you hear" to direct the large language model in performing the AAC task. As illustrated in Equation (1), the joint embedding E_J during training includes the text embedding E_T of the ground truth caption for the input audio, with training conducted using teacher forcing. Our system's training objective is the cross-entropy loss, defined as follows:

$$\mathcal{L}_{CE} = -\frac{1}{|E_T|} \sum_{t=1}^{|E_T|} \log p(E_{T,t} | E_{T,1:t-1}, E_J) \tag{2}$$

where $E_{T,t}$ represents the t-th token in the ground truth text embedding E_T, and $|E_T|$ denotes the length of the text embedding.

To optimize training cost and efficiency, we utilize parameter-efficient fine-tuning (PEFT) methods. Specifically, we employ LoRA [19] as an adapter, which is integrated into Vicuna to fine-tune the q projection and v projection layers within the Transformer blocks. Consequently, the trainable parameters are confined to the projector for modal alignment and the LoRA modules, while the remainder of the model remains frozen.

In the phase of decoding, traditional AAC systems often employ beam search [2] and sampling methods [1] for text generation. In our approach, however, we employ the CLAP model to filter generated descriptions, enhancing the alignment between the generated text and the input audio by leveraging multiple texts generated through various beam searches. CLAP [29] is a contrastive language-audio pre-training model that employs a feature fusion mechanism and keyword-to-caption augmentation to handle audio inputs of varying lengths. Trained on the extensive audio-text dataset LAION-Audio-630K, CLAP can encode text and audio data separately to obtain their respective representations E_A and E_T within a joint representation space. The cosine similarity between these embeddings is used to estimate the matching degree of text-audio pairs, as defined by the following similarity calculation:

$$\text{Similarity}(E_A, E_T) = \frac{E_A \cdot E_T}{\|E_A\| \|E_T\|} \tag{3}$$

In our decoding process, for the same input audio, we first use our system to generate the most probable sentences through different beam searches. Subsequently, the CLAP model calculates the cosine similarity scores between these generated captions and the input audio. The caption with the highest similarity score is then selected as the final caption for the audio.

2.2. Data Augmentation

To enhance our model's generalization capability in AAC tasks, we employ two primary data augmentation methods:

- For audio data, we utilize SpecAugment [30]. In our experiments, we apply masking to the mel-spectrogram of the audio, covering one-eighth of the range in the time dimension and one-quarter of the range in the frequency dimension.
- For audio-text pair data, some of our models employ the ChatGPT Mix-up Augmentation method [1]. This approach extends the pre-training Clotho dataset by using ChatGPT to combine captions from two different audio clips, thereby making the combined captions more natural and fluent. The corresponding audio waveforms are then mixed using the traditional mixup method [31] during training.

3. EXPERIMENTS

3.1. Datasets

In our experiments, the model was pre-trained using four main audio captioning datasets: Clotho [5], Audiocaps [4], Wavcaps [7], and MACS [6]. Specifically, we utilized the development set of Clotho, the training sets of Audiocaps and MACS, and the entire Wavcaps dataset for training.

Clotho v2 is sourced from the Freesound platform, featuring audio clips each lasting between 15 to 30 seconds. It is divided into several splits: the development split includes 3,839 audio clips with 19,195 captions; the validation split has 1,045 audio clips with 5,225 captions; the evaluation split also contains 1,045 audio clips with 5,225 captions; and the testing split comprises 1,043 audio clips with 5,215 captions. Each caption ranges from 8 to 20 words. These splits are created by constructing sets of unique words from each audio clip's captions.

Audiocaps is a large-scale audio captioning dataset with over 50k audio clips. Each audio clip in the dataset has a duration of 10 seconds and is sourced from the AudioSet dataset. Due to copyright constraints, the dataset we downloaded is divided into three subsets: training, validation, and testing, containing 49,274, 494, and 957 audio clips, respectively. In the training set, each audio clip is paired with a single caption, whereas the validation and test sets feature five captions per audio clip.

Wavcaps is a comprehensive audio captioning dataset comprising 403,050 audio clips, with an average text length of 7.80 words. The audio clips are sourced from the AudioSet Strongly Labeled Subset, BBC Sound Effects, FreeSound, and SoundBible

[2] EAT-base_epoch30 (fine-tuning on AS-2M)

[3] Vicuna-7b-v1.5

Table 1: Performance comparison of our systems and baseline model on the Clotho testing split. All metrics are reported such that higher values denote superior performance.

System	#Models	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr	SPIDEr-FL	FENSE
Baseline	1	0.1897	-	0.4619	0.1335	0.2977	0.2962	0.5040
Submission #1	1	0.1945	0.4003	0.5116	0.1470	0.3293	0.3290	**0.5431**
Submission #2	5	**0.1959**	**0.4060**	**0.5374**	**0.1495**	**0.3435**	**0.3417**	0.5398
Submission #3	10	0.1926	0.4035	0.5179	0.1476	0.3327	0.3315	0.5429
Submission #4	10	0.1933	0.4040	0.5216	0.1476	0.3346	0.3333	0.5429

websites. The dataset has been processed using ChatGPT to ensure high-quality annotations.

MACS dataset includes recordings from three acoustic scenes (airport, public square, and park) from the TAU Urban Acoustic Scenes 2019 dataset. Each of the 3,930 audio files is 10 seconds long and was annotated by 133 students, each providing annotations for up to 131 files. The annotations involved selecting sounds from a predefined list and writing free-form sentence descriptions. The dataset includes a total of 17,275 captions, with each audio file having 2 to 5 captions.

3.2. Training Details

Based on the EAT model's input audio processing, waveforms are resampled to 16 kHz and transformed into 128-dimensional mel-spectrograms utilizing a 25 ms Hanning window with a 10 ms shift. To enhance training efficiency, the audio duration is constrained to the initial 10 seconds for audio captioning.

During pre-training, we primarily utilize datasets including Clotho, Audiocaps, Wavcaps, and MACS. Additionally, some models incorporate the Clotho dataset augmented with the GPT Mix-up Augmentation method. We employ a batch size of 4 and a learning rate of 1e-4, conducting training for 100k updates. For fine-tuning on the Clotho development dataset, the batch size remains 4, while the learning rate is reduced to 8e-6. A linear learning rate schedule is implemented with a warmup of 1k updates, followed by linear decay. Model validation is performed every 500 updates, and checkpoints are saved based on the lowest validation loss.

During inference, our approach employs a beam search with beam widths ranging from 1 to 6 for each input audio, recording the highest probability captions as candidates. The final caption is determined using the CLAP model, which selects the caption with the highest similarity score.

The pre-training and fine-tuning processes were executed on an NVIDIA A800 GPU, requiring 26 hours and 5 hours, respectively.

3.3. Ensemble Method

To enhance the robustness of our system, our submission integrates results from an ensemble of multiple audio captioning models. Specifically, we modify the single-system beam search by averaging vocabulary probabilities from multiple systems at each decoding step. Additionally, we apply a length penalty method [32] to mitigate the risk of the ensemble model producing overly brief sentences during decoding.

4. RESULTS

In the DCASE 2024 Challenge, we submitted inference results of four systems on the Clotho testing set, comprising one single model system and three ensemble model systems. We trained multiple models with variations in random seeds, weight decay parameters, and the inclusion of ChatGPT Mix-up Augmenting data in the pre-training dataset. Based on the number of models combined, we created ensemble systems of 5 and 10 models.

For submission #1, we provided inference results from a single model. Submission #2 included results from an ensemble of 5 models. Submissions #3 and #4 consisted of results from an ensemble of 10 models, differing in the beam widths used for CLAP filtering during the decoding process. Specifically, submission #3 used candidates from beam widths 1 to 6, while submission #4 used candidates from beam widths 1 to 5.

Table 1 presents the performance of our systems compared to the baseline model on the Clotho testing set. The baseline model is a sequence-to-sequence system provided by DCASE, utilizing a frozen ConvNeXt as the audio encoder and a Transformer as the decoder. The results demonstrate that our models significantly outperform the baseline across various metrics. The ensemble models show notable improvements over the single model in ROUGE-L, CIDEr, SPICE, SPIDEr, and SPIDEr-FL metrics, while exhibiting slight declines or stability in METEOR and FENSE metrics.

5. CONCLUSION

This paper introduced our methods for submitting to the DCASE 2024 Challenge Task 6. We employ the self-supervised EAT model to extract audio features, subsequently aligning audio representations and text embeddings via lightweight linear projection layers. Decoding is performed using the large language model Vicuna. For efficient training, we fine-tuned only the projector and the LoRA modules inserted into the large language model. To enhance the alignment between generated captions and the input audio, the CLAP model was utilized to filter the captions. The results from our systems, submitted for the challenge, showed substantial improvements over baseline models across multiple performance metrics.

6. REFERENCES

[1] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. Le Roux, and S. Watanabe, "BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up," *Detection Classification Acoust. Scenes Events Challenge, Tech. Rep*, 2023.

[2] J.-H. Cho, Y.-A. Park, J. Kim, and J.-H. Chang, "Hyu submission for the dcase 2023 task 6a: Automated audio captioning model using al-mixgen and synonyms substitution," in *Proc. Detection and Classification of Acoustic Scenes and Events*, 2023.

[3] X. Xu, Z. Xie, M. Wu, and K. Yu, "The SJTU system for DCASE2021 challenge task 6: Audio captioning based on encoder pre-training and reinforcement learning," *DCASE2021 Challenge, Tech. Rep, Tech. Rep*, 2021.

[4] 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)*, 2019, pp. 119–132.

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

[6] I. Martin Morato and A. Mesaros, "Diversity and bias in audio captioning datasets," 2021.

[7] 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.

[8] 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.

[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] 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)*. IEEE, 2022, pp. 981–985.

[11] 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.

[12] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[13] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[14] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[15] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," *arXiv preprint arXiv:2309.05519*, 2023.

[16] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[17] W. Chen, Y. Liang, Z. Ma, Z. Zheng, and X. Chen, "EAT: Self-Supervised Pre-Training with Efficient Audio Transformer," *arXiv preprint arXiv:2401.03497*, 2024.

[18] J. F. Gemmeke, D. P. 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)*. IEEE, 2017, pp. 776–780.

[19] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," 2021.

[20] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," *Advances in neural information processing systems*, vol. 33, pp. 21 271–21 284, 2020.

[21] 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.

[22] P.-Y. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28 708–28 720, 2022.

[23] 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.

[24] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[25] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 1015–1018.

[26] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[27] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang, *et al.*, "An Embarrassingly Simple Approach for LLM with Strong ASR Capacity," *arXiv preprint arXiv:2402.08846*, 2024.

[28] Z. Zheng, P. Peng, Z. Ma, X. Chen, E. Choi, and D. Harwath, "BAT: Learning to Reason about Spatial Sounds with Large Language Models," *arXiv preprint arXiv:2402.01591*, 2024.

[29] 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 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[30] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[31] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[32] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.