THE SJTU SYSTEM FOR DCASE2022 CHALLENGE TASK 6: AUDIO CAPTIONING WITH AUDIO-TEXT RETRIEVAL PRE-TRAINING

Technical Report

Xuenan Xu, Zeyu Xie, Mengyue Wu, Kai Yu

MoE Key Lab of Artificial Intelligence
X-LANCE Lab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
{*wsntxxn, mengyuewu, kai.yu*}@sjtu.edu.cn, zeyuxie29@gmail.com

ABSTRACT

This technical report describes the system submitted to the Detection and Classification of Acoustic Scenes and Events (DCASE) 2022 challenge Task 6. There are two involving subtasks: text-to-audio retrieval and automated audio captioning. The text-to-audio retrieval system adopts a bi-encoder architecture using pre-trained audio and text encoders. The system is first pre-trained on AudioCaps and then fine-tuned on the challenge dataset Clotho. For the audio captioning system, we first train a retrieval model on all public captioning data and then take the audio encoder as the feature extractor. Then a standard sequence-to-sequence model is trained on Clotho based on the pre-trained feature extractor. The captioning model is first trained by word-level cross entropy loss and then fine-tuned using self-critical sequence training. Our system achieves a SPIDEr of 32.5 on captioning and an mAP of 29.9 on text-to-audio retrieval.

Index Terms— Audio captioning, text-to-audio retrieval, contrastive learning, pre-training

1. INTRODUCTION

Automated audio captioning and audio-text retrieval are cross-modal tasks involving both audio recognition and natural language processing. Audio captioning aims to generate audio content description using free text while audio-text retrieval searches the corresponding audio given the text query or vice versa. They can be helpful in applications like automatic content description and multimedia search. The interaction between audio and text has attracted much attention from researchers recently [1, 2].

Over the last few years, automated audio captioning has overseen rapid development thanks to its inclusion in previous DCASE challenges, with moving progress in data, model architecture, training schemes and evaluation metrics (for a full review see [3]). For the current audio captioning system, we employ a standard sequence-to-sequence model, consisting of an audio encoder and a text decoder. We take features extracted by pre-trained models as the input to the captioning model to reduce the number of trainable parameters. Compared with models like PANNs or AST [4], the audio encoder trained by audio-text retrieval learns to differentiate textual descriptions, which makes it better at extracting caption-related audio features. Therefore, we take the audio encoder of the pre-trained retrieval model as the feature extractor. Following previous works [5, 6], we further fine-tune the captioning model by directly optimizing the evaluation metric CIDEr using self-critical sequence training (SCST) [7].

Natural language-based audio-text retrieval is first proposed in [8] using Mixture-of-Embedded Experts (MoEE) [9] and Collaborative-Experts (CE) [10] frameworks. [11] compares different audio features and aggregation methods and finds that PANNs [12] features with NetRVLAD [13] aggregation perform the best. In their approaches the pre-trained audio classification model is frozen and used as the audio feature extractor. The text feature extractor is simply a pre-trained Word2vec. However, due to the annotation errors in pre-training data [14] and domain mismatch, fine-tuning the audio and text feature extractors can be expected to boost performance. To better leverage the pre-trained models in retrieval, we incorporate the feature extractors into the training process. The retrieval system utilizes a bi-encoder architecture where the input audio and text are encoded separately. All parameters in the model are trainable. We explore several pre-trained large models for audio and text encoders, including deep convolution neural networks (CNN) and Transformers. We use a simple linear projection layer to map audio and text embeddings into the same embedding space to reduce the parameter number. Different from previous works, the whole retrieval model is trained by InfoNCE loss [15] due to its superior performance in contrastive learning. The ensemble of different architectures further improves the retrieval performance significantly.

The remaining of this report is organized as follows. Section 2 describes our framework and architecture. The experimental setup is given in Section 3. Section 4 presents the results on the public evaluation set. Finally, conclusion is drawn in Section 5.

2. SYSTEM DESCRIPTION

2.1. Audio Captioning

The audio captioning system consists of a feature extractor Etr, an audio encoder Enc and a text decoder Dec. The feature extractor transforms the input audio clip $\mathcal{A}$ into audio feature $\mathbf{E}_{\mathcal{A}}$. Then Enc further extracts audio embedding $\mathbf{E}$ based on $\mathbf{E}_{\mathcal{A}}$.

$$\begin{aligned} \mathbf{E}_{\mathcal{A}} &= \mathrm{Etr}(\mathcal{A}) \\ \mathbf{E} &= \mathrm{Enc}(\mathbf{E}_{\mathcal{A}}) \end{aligned} \tag{1}$$

The Dec accepts the previous words and $\mathbf{E}$ as the input. A fully connected classifier outputs the word probability based on the Dec

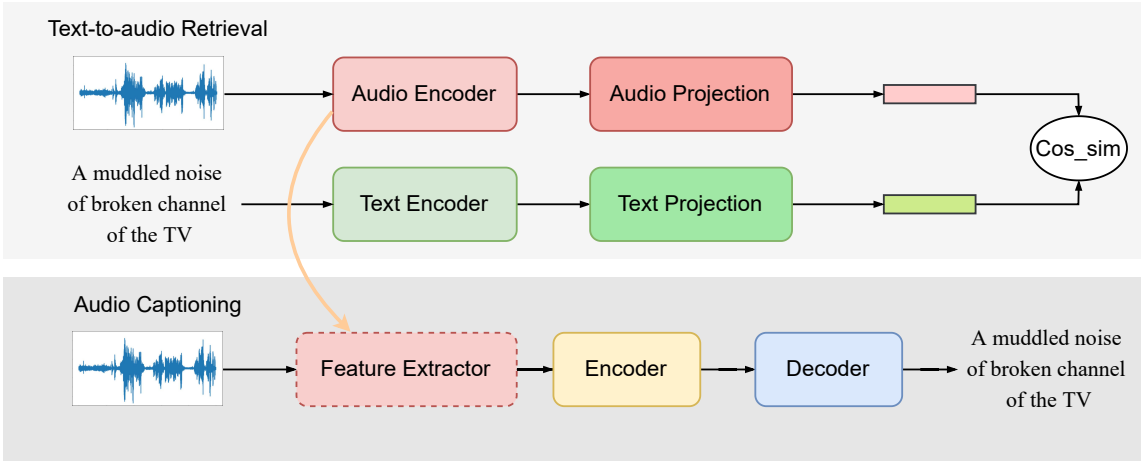

Figure 1: An overview of our approach. The audio-text retrieval model is first trained on parallel audio-text data. Then the audio encoder is used as the feature extractor (frozen) of the captioning system. "Cos_sim" denotes cosine similarity.

Enc	BiGRU (256)×3		
Dec	Transformer (256)×2	Transformer (256)×4	GRU (512)×1
# param / M	10.4	12.5	14.6

Table 1: Architectures and parameter numbers of captioning models. The number in the brackets indicates hidden size while the number following "×" indicates the number of layers.

output.

$$\begin{aligned} \mathbf{y} &= \text{Dec}(\mathbf{E}, \text{WE}(\mathbf{w}_{previous})) \\ \mathbf{o} &= \text{Classifier}(\mathbf{y}) \end{aligned} \tag{2}$$

where the word embedding layer WE maps words into the embedding space.

We adopt a pre-trained Etr which will be discussed in 2.3. A three-layer bidirectional gated recurrent unit (GRU) is taken as Enc. We use two kinds of models as Dec, GRU with attention mechanism and Transformer. The architectures are shown in Table 1.

The entire model except for the frozen Etr is trained with cross entropy (XE) loss between the estimated word probability p and ground truth word w_t:

$$\mathcal{L}_{\text{XE}} = -\frac{1}{T}\sum_{t=1}^{T} \log p(w_t) \tag{3}$$

After XE training, the model is further fine-tuned using reinforcement learning by SCST to optimize the evaluation metric CIDEr.

2.2. Text-to-audio Retrieval

The text-to-audio retrieval system is a bi-encoder model, consisting of an audio encoder Enc_A, a text encoder Enc_T and a cross-modal matching module. For an input audio-text pair $(\mathcal{A}, \mathcal{T})$, the two encoders transform the audio and text into their embeddings $\mathbf{a}$ and $\mathbf{t}$ respectively.

$$\begin{aligned} \mathbf{a} &= \text{Enc}_A(\mathcal{A}) \\ \mathbf{t} &= \text{Enc}_T(\mathcal{T}) \end{aligned} \tag{4}$$

Then the matching module projects $\mathbf{a}$ and $\mathbf{t}$ into a common embedding space and calculates their similarity score:

$$\begin{aligned} \mathbf{a}_p &= \text{Proj}_A(\mathbf{a}) \\ \mathbf{t}_p &= \text{Proj}_T(\mathbf{t}) \\ s &= \frac{\mathbf{a}_p \cdot \mathbf{t}_p^{\mathrm{T}}}{\|\mathbf{a}_p\| \cdot \|\mathbf{t}_p\|} \end{aligned} \tag{5}$$

where Proj_A and Proj_T are projection layers. We use a single fully-connected (FC) layer as the projection layer. Cosine similarity is used as the similarity metric. The training loss is the widely-adopted InfoNCE loss [15] for its effectiveness in differentiating confusing samples. In a batch containing N audio-text pairs $(\mathcal{A}_1, \mathcal{T}_1), (\mathcal{A}_2, \mathcal{T}_2), \ldots, (\mathcal{A}_N, \mathcal{T}_N)$, the model calculates the pairwise audio-text similarity $s(i, j)$. The cross entropy loss between the similarity scores and the ground truth labels is calculated as the contrastive training loss:

$$\begin{aligned} \mathcal{L}_i^{A\to T} &= -\log \frac{exp\left(s(i,i)/\tau\right)}{\sum_{j=1}^{N} exp(s\left(i,j\right)/\tau)} \\ \mathcal{L}_i^{T\to A} &= -\log \frac{exp\left(s(i,i)/\tau\right)}{\sum_{j=1}^{N} exp(s\left(j,i\right)/\tau)} \\ \mathcal{L} &= \frac{1}{N}\sum_{i=1}^{N}(\mathcal{L}_i^{A\to T} + \mathcal{L}_i^{T\to A}) \end{aligned} \tag{6}$$

where τ is the trainable temperature. The audio-to-text cross entropy loss is included to enhance the model ability of aligning audio and text.

In this framework, the audio encoder and the text encoder can adopt different architectures. We explore several frequently used audio classification and natural language understanding models, including CNN14 and Wavegram-Logmel-CNN14 in PANNs, AST[1]

[1]To reduce the memory required for training, we use a stride of 16 for patch splitting on both the time and frequency dimension

Enc_A	Enc_T	# param / M
WLCNN14 [12]	$\text{BERT}_{\text{BASE}}$ [16]	192
CNN14 [12]	$\text{BERT}_{\text{MEDIUM}}$ [18]	124
CNN14	$\text{BERT}_{\text{BASE}}$	192
CNN14	$\text{RoBERTa}_{\text{BASE}}$ [17]	207
AST [4]	$\text{BERT}_{\text{BASE}}$	196

Table 2: Architectures and parameter numbers of retrieval models.

for Enc_A, BERT [16] and RoBERTa [17] for Enc_T. The architectures are listed in Table 2.

2.3. Pre-training

In previous works, the audio encoder or feature extractor of the captioning model are mostly pre-trained by audio event recognition. To enable the model to better encode information for caption generation, we use audio-text matching (retrieval) task for pre-training. The audio encoder part of the retrieval model is trained to capture text-related feature by contrastive training. Then the audio encoder is used as the feature extractor of the captioning model. It is frozen during captioning training. The procedure is shown in Figure 1. We use current public audio captioning datasets, including Clotho [19], AudioCaps [20] and MACS [21] for pre-training.

For text-to-audio retrieval, we also pre-train the model first. However, we find that the model pre-trained on the mixture of all captioning data does not perform as well as that pre-trained on AudioCaps only. Therefore, we pre-train the retrieval model on AudioCaps and then fine-tune it on Clotho.

3. EXPERIMENT

Data For both subtasks, Clotho v2.1 is used as the dataset, with 3839, 1045 and 1045 audio clips in the training, validation and test sets. We merge the original training and validation sets. Then the merged development set is split into new training and validation subsets in a $9:1$ ratio. The re-splitting is used to get more data for training.

Text-to-audio Retrieval In both pre-training and fine-tuning, the retrieval model is trained for 20 epochs with a batch size of 128. The audio encoder and text encoder parameters are all initialized with pre-trained ones so a smaller learning rate is used. We use the Adam optimizer The learning rate linearly warms up in the first epoch and then decayed by a cosine scheduler. The maximum learning rate during pre-training is 1×10^{-4} while during fine-tuning is 2×10^{-5}.

After training the model with different audio encoder and text encoder architectures, we ensemble these models for better performance. The ensemble setup of four submissions is given below:

- system 1: Ensemble of all five models.
- system 2: Ensemble of the first four models.
- system 3: Ensemble of the first, third and fourth model.
- system 4: Ensemble of the first three models.

Audio Captioning We first train an audio-text retrieval model on the mixture of public audio captioning datasets (Clotho, AudioCaps, MACS) with the same configuration as the pre-training stage mentioned above. Then the audio encoder of the retrieval model is taken as the feature extractor of the captioning model.

The whole captioning model except the feature extractor is trained for 25 epochs with a batch size of 64 using the Adam optimizer. Label smoothing ($\alpha = 0.1$) is used to prevent over-fitting. The learning rate increases linearly to 5×10^{-4} in first 3000 warm-up iterations, and then decays exponentially to 5×10^{-7} at the end of training. Schedule sampling is used with the probability of teacher forcing decreasing linearly from 1 to 0.7. After XE training, the model is fine-tuned by SCST for 100 epochs with a learning rate of 5×10^{-5}. During inference, beam search with a size of 3 is used. Different models are ensembled to further enhance the performance. Here are our submission setups:

- system 1: SCST fine-tuned model using Transformer decoder (two layers).
- system 2: Ensemble of two system 1.
- system 3: Ensemble of system 2 and two SCST fine-tuned models using Transformer decoder (four layers).
- system 4: Ensemble of system 3 and two SCST fine-tuned models using attentional GRU decoder.

4. RESULTS

4.1. Audio captioning

The performance of our audio captioning systems is presented in Table 3. Whether XE loss training or SCST fine-tuning, the 2 layers Transformer text decoder model shows weak advantages over the 4 layers Transformer text decoder model and GRU based decoder model. SCST fine-tuning enhances the performance in all models, especially in ROUGE_L and objective metrics CIDEr, which brings the highest SPIDEr score 31.5 of a single model. Ensemble integrates outputs from multiple models. The highest SPIDEr (32.5) is achieved by the ensemble of two models using Transformer decoder initialized with different random seeds. The ensemble of different architectures (Transformer + GRU) achieve similar results with that from the same architectures.

4.2. Text-to-audio Retrieval

The text-to-audio retrieval performance is shown in Table 4. Wavegram-Logmel-CNN14 takes both the raw waveform and log-mel spectrogram into audio encoding and presents better performance than CNN14 and AST. AST encoder performs the worst, possible due to the mismatch in data pre-processing (we use larger patch splitting strides than that used during pre-training on AudioSet). With the same audio encoder, different text encoders achieve similar performance while RoBERTa performs slightly better than the original BERT. Although models with different architectures achieve similar results, the ensemble of them improves the retrieval performance significantly. We also ensemble results from five WLCNN14-BERT_B models, which are initialized from different random seeds. However, it only achieves an mAP@10 of 25.9. Compared with audio captioning, the ensemble of different architectures is much more effective than the ensemble of the same architecture.

	B@4	R	M	C	S	SD	B@4	R	M	C	S	SD
Decoder	**XE training**						**CIDEr-D Optimization**					
one-layer GRU (Attention)	16.4	38.6	18.1	42.1	12.6	27.4	17.2	40.9	18.1	48.7	12.1	30.4
two-layer Transformer	16.6	37.8	17.9	42.1	12.7	27.4	18.2	41.2	18.6	50.9	12.0	31.5
four-layer Transformer	16.0	37.9	18.1	41.1	12.5	26.8	17.4	41.3	18.0	50.0	12.2	31.1
System	**Ensemble**											
two-layer Transformer $\times$ 2	17.4	38.5	18.3	43.2	13.0	28.1	18.7	41.3	18.8	52.4	12.6	32.5
$\searrow$ +four-layer Transformer $\times$ 2	17.6	39.0	18.4	43.8	13.1	28.5	18.3	41.5	18.6	51.3	12.6	32.0
$\searrow$ +GRU $\times$ 2	17.4	39.4	18.5	43.9	13.2	28.5	18.3	41.5	18.6	51.3	12.6	32.0

Table 3: The performance of systems with different text decoders and ensemble strategies. In ensemble strategy, "$\times 2$" indicates that two different random seeds are used and then ensembled. "$\searrow$ +" denote that the system in the above line is ensembled with the following system together. B@4, R, M, C, S and SD denote BLEU_4, ROUGE_L, METEOR, CIDEr, SPICE and SPIDEr, respectively.

Enc_A	Enc_T	R@1	R@5	R@10	mAP@10
WLCNN14	BERT_B	15.4	38.9	53.0	25.6
CNN14	BERT_M	15.0	38.2	52.0	25.1
CNN14	BERT_B	14.9	38.1	52.3	24.9
CNN14	RoBERTa_B	16.2	38.3	52.0	25.8
AST	BERT_B	14.6	37.6	50.2	24.4
Ensemble		18.8	44.7	58.7	29.9

Table 4: The performance of systems with different audio encoder and text encoder architectures on Clotho evaluation set. "WLCNN14" denotes Wavegram-Logmel-CNN14 and the subscript "M" and "B" denote MEDIUM and BASE respectively.

5. CONCLUSION

In this report, we comprehensively described our system for DCASE2020 challenge Task 6. For text-to-audio retrieval, we train a bi-encoder model with pre-trained audio and text encoders using InfoNCE loss. The model is first pre-trained on AudioCaps and then fine-tuned on Clotho. For audio captioning, we take the audio encoder of the pre-trained retrieval model as the feature extractor and train a sequence-to-sequence model. The captioning model is further fine-tuned by SCST. Ensemble of several models improve the performance of both captioning and retrieval significantly.

6. REFERENCES

[1] Y. Zhao, J. Hessel, Y. Yu, X. Lu, R. Zellers, and Y. Choi, "Connecting the dots between audio and text without parallel data through visual knowledge transfer," *arXiv preprint arXiv:2112.08995*, 2021.

[2] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," *arXiv preprint arXiv:2106.13043*, 2021.

[3] X. Xu, M. Wu, and K. Yu, "A comprehensive survey of automated audio captioning," *arXiv preprint arXiv:2205.05357*, 2022.

[4] G. Yuan, C. Yu-An, and G. James, "AST: Audio Spectrogram Transformer," in *Proceedings of Conference of the International Speech Communication Association*, 2021, pp. 571–575.

[5] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. ZHAO, S. Li, T. Ko, H. Tang, X. Shao, M. D. Plumbley, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 206–210.

[6] Z. Ye, H. Wang, D. Yang, and Y. Zou, "Improving the performance of automated audio captioning via integrating the acoustic and semantic information," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 40–44.

[7] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7008–7024.

[8] A.-M. Oncescu, A. S. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio Retrieval with Natural Language Queries," in *Proceedings of Conference of the International Speech Communication Association*, 2021, pp. 2411–2415.

[9] A. Miech, I. Laptev, and J. Sivic, "Learning a text-video embedding from incomplete and heterogeneous data," *arXiv preprint arXiv:1804.02516*, 2018.

[10] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman, "Use what you have: Video retrieval using representations from collaborative experts," in *arXiv preprint arxiv:1907.13487*, 2019.

[11] S. Lou, X. Xu, M. Wu, and K. Yu, "Audio-text retrieval in context," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4793–4797.

[12] 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.

[13] A. Miech, I. Laptev, and J. Sivic, "Learnable pooling with context gating for video classification," *arXiv preprint arXiv:1706.06905*, 2017.

[14] Y. Gong, Y.-A. Chung, and J. Glass, "Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 29, pp. 3292–3306, 2021.

[15] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[16] 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.

[17] 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.

[18] I. Turc, M.-W. Chang, K. Lee, and K. Toutanova, "Well-read students learn better: On the importance of pre-training compact models," *arXiv preprint arXiv:1908.08962*, 2019.

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

[20] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019, pp. 119–132.

[21] I. Martin and A. Mesaros, "Diversity and bias in audio captioning datasets," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Barcelona, Spain, November 2021, pp. 90–94.