THE SJTU SYSTEM FOR DCASE2021 CHALLENGE TASK 6: AUDIO CAPTIONING BASED ON ENCODER PRE-TRAINING AND REINFORCEMENT LEARNING

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 report proposes an audio captioning system for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2021 challenge task Task 6. Our audio captioning system consists of a 10-layer convolution neural network (CNN) encoder and a temporal attentional single layer gated recurrent unit (GRU) decoder. In this challenge, there is no restriction on the usage of external data and pre-trained models. To better model the concepts in an audio clip, we pre-train the CNN encoder with audio tagging on AudioSet. After standard cross entropy based training, we further fine-tune the model with reinforcement learning to directly optimize the evaluation metric. Experiments show that our proposed system achieves a SPIDEr of 28.6 on the public evaluation split without ensemble[1].

Index Terms— Audio captioning, pre-training encoder, reinforcement learning, audio tagging

1. INTRODUCTION

Automated audio captioning (AAC) is an intermodal translation task defined as generating the textual description for an input audio clip [1]. It has various potential applications for high level understanding of the acoustic environment, e.g., automatic content summarization, intelligent human-machine interaction. The introduction to AAC in the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 challenge has attracted much more attention recently [2, 3, 4, 5].

Different levels of concepts are modeled in AAC, including acoustic scenes (e.g. ”street”), sound events (e.g. ”car horn”), physical properties of events (e.g. ”a **wooden** door”), and high level knowledge (”a clock rings **three times**”). However, it is difficult for the encoder to model acoustic scenes and events via training from scratch due to the limitation of the dataset size and the indeterminacy problem of the supervision signal, i.e., caption. To make the encoder eligible to recognize discriminative audio patterns, pre-training and multi-task learning are proposed [6, 7]. Since any external data or pre-trained models are allowed in DCASE2021 challenge Task 6, we incorporate the encoder pre-training before the standard training procedure. Pre-training the encoder by a sound classification task (e.g., audio tagging) significantly improves its ability to encode concepts in audio. In addition, previous works also show that reinforcement learning (RL) based on policy gradient can estimate the parameter gradients when there are non-differentiable operations, landing on directly optimizing the evaluation metrics [8]. Therefore we combine the standard XE training with encoder pre-training and RL fine-tuning to improve the captioning performance.

The remainder of this report is structured as follows. Section 2 introduces our system. Section 3 describes the detailed implementation. Section 4 presents our results on the public evaluation set. Section 5 concludes our work.

2. SYSTEM DESCRIPTION

In this section, we describe the details of our system, including our proposed training procedure, the architecture of our sequence-to-sequence model, the data augmentation and regulation strategies.

2.1. Training Procedure

Our audio captioning model consists of an audio encoder and a text decoder. Since encoder pre-training and reinforcement learning based fine-tuning effectively boost performance, our proposed training procedure consists of three stages: 1) encoder pre-training; 2) standard cross entropy (XE) based training; 3) RL based fine-tuning.

Encoder Pre-training The goal of encoder pre-training is to improve the ability of the encoder to extract embeddings containing patterns of the input audio, e.g. acoustic scenes and events. We pre-train the encoder on the large-scale acoustic event dataset AudioSet [9]. The pre-training task is audio tagging, aiming at detecting all the pre-defined E acoustic events present in an input audio. Training is done by minimizing the binary cross entropy (BCE) loss between the output event probability $p(e)$ and the event label $y(e)$:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{E}\sum_{e=1}^{E} p(e)\log(y(e)) + (1-p(e))\log(1-y(e)) \quad (1)$$

here $E = 527$ in AudioSet, $y(e) = 1$ for event e presented in the audio and 0 for other events.

XE Training After the encoder pre-training, the whole model is trained by the standard XE loss between the estimated word probability p and the annotation given the input audio feature $\mathbf{X}$:

$$\mathcal{L}_{\text{XE}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \log p(w_t^*|\theta) \quad (2)$$

[1] The code and models are available at `https://github.com/wsntxxn/AudioCaption`

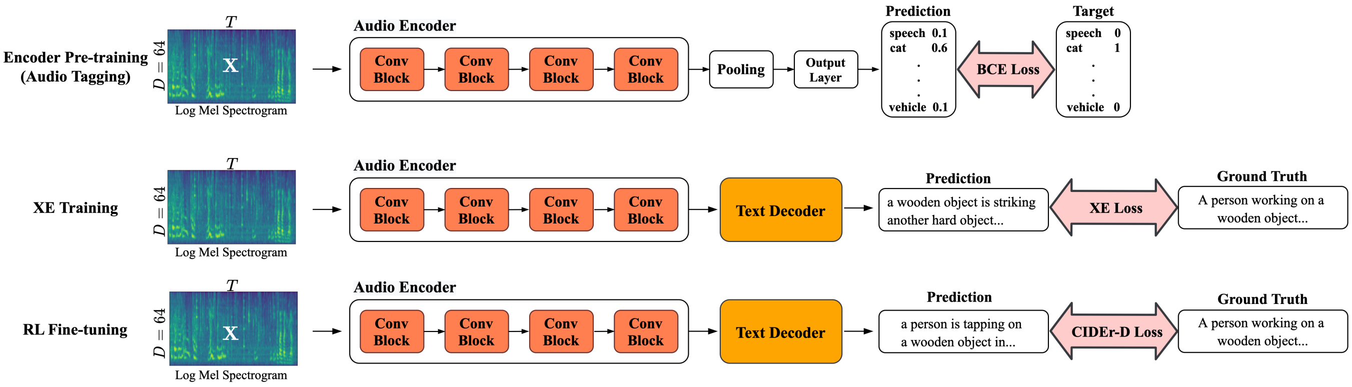

Figure 1: Our proposed training procedure. The encoder pre-training is first done by audio tagging. Then the convolution layers are used to initialize the encoder in the caption model. RL fine-tuning is applied after XE training to directly optimize the metrics.

where $s^* = (w_1^*, w_2^*, \dots, w_T^*)$ denotes the ground truth sentence containing T words.

RL Fine-tuning After the model is trained by XE loss for certain epochs, we conduct reinforcement learning to directly optimize the evaluation metric. The XE trained model is used to initialize the parameters to ensure that the model can get enough reward for training at the beginning. The training objective is to minimize the negative reward of the sampled sentence:

$$\mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{s^s \sim p(s)}[r(s^s)] \tag{3}$$

where $s^s = (w_1^s, w_2^s, \dots, w_T^s)$ and w_t^s denotes the word sampled given the output probability at the time step t. Then the parameter gradients can be computed by policy gradient:

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta) = -\mathbb{E}_{s^s \sim p(s)}[r(s^s) \nabla_\theta \log p(s^s)] \tag{4}$$

Following the previous work [10] we estimate the gradients by a single Monte-Carlo sample and use the greedy decoding reward as the reference baseline:

$$\nabla_\theta \mathcal{L}_{\mathrm{RL}}(\theta) \approx -[r(s^s) - r(\hat{s})] \nabla_\theta \log p(s^s) \tag{5}$$

here $\hat{s}$ is the greedy decoding sentence. CIDEr (specifically CIDEr-D) [11] is chosen as the optimization objective for its fast computation speed.

2.2. Model Architecture

Our audio captioning model utilizes a standard sequence-to-sequence architecture. An audio encoder first extracts an abstract embedding sequence from the input audio. Then a text decoder translates the sequence into the caption description. The backbone architecture is a temporal attentional convolution neural network (CNN) - gated recurrent unit (GRU), which can be found in [12].

Audio Encoder We use a 10-layer CNN (CNN10) as our backbone audio encoder for its success in audio pattern recognition [13], including previous audio captioning works [6, 14]. It consists of an initial batch normalization layer, four convolution blocks and two fully connected layers. Each convolution block contains two convolution layers with 3 × 3 kernels, two batch normalization layers with ReLU activation. All convolutions use zero padding to preserve the input size. 2 × 2 average pooling and dropout with 0.2 ratio are applied after each convolution block. It should be noted that after the encoder pre-training, the last two fully connected layers are dropped while only the convolution layers are used for XE training initialization.

Text Decoder We use a single layer GRU with temporal attention mechanism as the text decoder. Each word is embedded to a 512-dimension vector and apply a dropout operation with ratio 0.5 before feeding to the GRU. At each decoding time step, the decoder accepts both the previous word embedding and the context vector as the input. The context vector is a weighted combination of the embedding sequence, which is calculated by attention mechanism. The previous hidden state is taken as query while the embedding sequence is taken as both the key and value. A fully connected classifier outputs the word probability based on the GRU output. The decoding step iterates until " <EOS> " (see Section 3) or the maximum caption length (20) is reached.

2.3. Data Augmentation and Regulation

To avoid over-fitting to specific patterns, we apply SpecAugment [15] before feeding the input feature to the model. Frequency channel and time step blocks are randomly masked. Label smoothing [16] is applied to smooth the one-hot word label. We also apply scheduled sampling to gently decrease the training-inference discrepancy caused by teacher forcing training [17]. The probability of feeding the ground truth word to the decoder is linearly decayed from 1.0 to 0.7.

3. EXPERIMENTAL SETUP

Clotho v2 is used in this challenge, including three subsets: development, validation and evaluation. They contain 3839, 1045 and 1045 audio clips, respectively. Each audio clip is annotated by five sentences. During training, each of the sentence is combined with the audio to form a training sample. In order to use more data for training, we merge the development and validation set and randomly split it into two new subsets (training and validation) in a $9:1$ ratio.

In data pre-processing, 64-dimensional log-Mel spectrogram (LMS) is extracted from audio as the input feature with a frame shift of 20 ms and a 40 ms Hann window. Two frequency masks and two time masks are used in SpecAugment with parameters W = 40, T = 30, probability p = 0.2. For each caption we remove punctuations

Table 1: Experimental results on the public Clotho evaluation set. B@1, B@4, M, R, C, S and SD denote BLEU_1, BLEU_4, METEOR, ROUGE_L, CIDEr, SPICE and SPIDEr, respectively. For all metrics, higher values indicate better performance.

	XE Training							**RL Fine-tuning**						
Model	B@1	B@4	M	R	C	S	SD	B@1	B@4	M	R	C	S	SD
								Single Model						
Baseline	37.8	1.7	7.8	26.3	7.5	2.8	5.1	-	-	-	-	-	-	-
Proposed	56.5	15.5	17.4	37.4	39.9	11.9	25.9	64.0	16.3	17.8	40.4	44.9	12.3	28.6
								Ensemble						
Ensemble	56.9	16.4	17.8	37.8	41.4	12.1	26.8	**65.7**	**17.4**	**18.2**	**40.8**	**46.8**	**12.3**	**29.5**

and convert all letters to lowercase to reduce the vocabulary size. Special tokens " <BOS> " and " <EOS> " to captions are added to mark the beginning and the end of sentences.

The encoder is first pre-trained on AudioSet unbalanced training set for at most 15 epochs with an early stopping strategy of 5 epochs. Then XE training takes 25 epochs, and RL fine-tuning takes 100 epochs. The model with the best performance on the validation set is chosen for initializing the next training stage (XE training or RL fine-tuning) or evaluation.

The initial learning rates and batch sizes are set to 10^{-3}, 5×10^{-4}, 5×10^{-5} and 64, 32, 32 for the three stages, respectively. During XE training, we automatically anneal the learning rate by multiplying it by 0.1 once the validation performance improve for five consecutive epochs. The label smoothing factor is set to 0.1. During evaluation and inference, we use beam search with a beam size of 3.

For submission, we adopt the ensemble strategy to further enhance the model performance. At each decoding time step t, the incorporated models $\theta_1, \theta_2, \ldots, \theta_n$ give the word probability $p(w|\theta_1), p(w|\theta_2), \ldots, p(w|\theta_n)$ respectively. The average of all probabilities are used for beam search. The decoded word is fed to all the models as the input of the next time step $t + 1$. Our submission systems are all ensembles. Each of them is ensembled by models trained with the same configurations but different random seeds. Here are our submission setups:

- Submission 1. Ensemble of three XE trained models with the highest CIDEr score.
- Submission 2. Ensemble of five XE trained models.
- Submission 3. Ensemble of four RL fine-tuned models with the highest CIDEr score.
- Submission 4. Ensemble of five RL fine-tuned models.

4. RESULTS

The performance of our proposed system is presented in Table 1. In addition to the single model, we also present the result of an ensemble of three models for comparison. Note that in the challenge submission we ensemble more models. Our proposed single model contains 12 million parameters in total. The results show that our proposed temporal attentional CNN10-GRU with encoder pretraining significantly outperforms the baseline. Though RL fine-tuning aims at optimizing CIDERr-D score, results indicate that such a training strategy further enhances the performance on all metrics, producing a SPIDEr as high as 28.6. The ensembled model reaches the summit SPIDEr of 29.5.

5. CONCLUSION

In this report, we describe our system submitted to DCASE2021 challenge Task 6. Our system is a standard sequence-to-sequence model including a 10-layer CNN and a single layer GRU with temporal attention mechanism. To enhance the encoder ability to recognize audio patterns, we pre-train the encoder on AudioSet before standard XE training. We further fine-tune the model by reinforcement learning to optimize CIDEr-D score. Our proposed nonensemble system achieves a SPIDEr of 28.6 on the public Clotho evaluation split.

6. 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)*. IEEE, 2017, pp. 374–378.

[2] M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 830–834.

[3] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A Transformer-based Audio Captioning Model with Keyword Estimation," in *Proc. ISCA Interspeech*, 2020, pp. 2–6. [Online]. Available: http://arxiv.org/abs/2007.00222

[4] E. Cakır, K. Drossos, and T. Virtanen, "Multi-task regularization based on infrequent classes for audio captioning," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Tokyo, Japan, November 2020, pp. 6–10.

[5] A. Ö. Eren and M. Sert, "Audio Captioning Based on Combined Audio and Semantic Embeddings," in *Proceedings of IEEE International Symposium on Multimedia (ISM)*, 2020.

[6] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pre-trained cnn," in *Proc. Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Tokyo, Japan, November 2020, pp. 21–25.

[7] Y. Koizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "The ntt dcase2020 challenge task 6 system: Automated audio captioning with keywords and sentence length estimation," DCASE2020 Challenge, Tech. Rep., June 2020.

[8] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

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

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

[11] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 07-12-June, 2015, pp. 4566–4575.

[12] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 905–909.

[13] 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.

[14] H. Wang, B. Yang, Y. Zou, and D. Chong, "Automated audio captioning with temporal attention," DCASE2020 Challenge, Tech. Rep., June 2020.

[15] 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," *Proc. ISCA Interspeech*, pp. 2613–2617, 2019.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[17] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Conference on Neural Information Processing Systems (NIPS)*, 2015, pp. 1171–1179.