AUTOMATED AUDIO CAPTIONING WITH TEMPORAL ATTENTION

Technical Report

Helin Wang[1], *Bang Yang*[1], *Yuexian Zou*[1,2,*], *Dading Chong*[1]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China

ABSTRACT

This technical report describes the ADSPLAB team's submission for Task6 of DCASE2020 challenge (automated audio captioning). Our audio captioning system is based on the sequence-to-sequence model. Convolutional neural network (CNN) is used as the encoder and a long-short term memory (LSTM)-based decoder with temporal attention is used to generate the captions. No extra data or pre-trained models are employed and no extra annotations are used. The experimental results show that our system could achieve the SPIDEr of 0.172 (official baseline: 0.054) on the evaluation split of the Clotho dataset.

Index Terms— Automated audio captioning, sequence-to-sequence model, temporal attention

1. INTRODUCTION

The automated audio captioning problem is defined as the task of automatically generating a textual description (*i.e.* caption) for an audio signal, where the caption is as close as possible to a human-assigned one [1]. Different from the sound event detection (SED) and audio tagging (AT) tasks, the audio captioning method does not predict sound events and their start and end times nor assigns labels to an audio file. Instead, more information needs to be described including the identification of sound events, acoustic scenes, spatiotemporal relationships of sources, foreground versus background discrimination, concepts, and physical properties of objects and environment [2]. Audio captioning can be used in various applications, such as intelligent and content oriented machine-to-machine interaction and automatic content description.

Several datasets were published for training an audio captioning system, including Audio Caption [3], AudioCaps [4] and Clotho [2]. Detection and Classification of Acoustic Scenes and Events (DCASE) challenges organized by IEEE Audio and Signal Processing (AASP) Technical Committee published the automated audio captioning task in DCASE2020 Task6 [5]. The Clotho dataset is used as the development and evaluation set, with totally 4981 audio samples and 24905 captions.

The report describes the details of ADSPLAB team's submission for Task6 of DCASE2020. Our system is a sequence-to-sequence model, which contains an encoder based on convolutional neural network (CNN) and a decoder based on long-short term memory (LSTM). In addition, temporal attention is applied for each time step of LSTM to focus on different temporal frames of the audio. On the official evaluation split of Clotho, our system could achieve the SPIDEr of 0.172.

The remainder of this report is organized as follows. Section 2 describes the architecture of our system. Section 3 presents the details of experiments and results. Section 4 concludes our work.

2. SYSTEM ARCHITECTURE

In this section, the architecture of our system is introduced, which is shown in Figure 1. Specifically, our system is based on a sequence-to-sequence model, including the down-sampling module, the CNN encoder module and the LSTM-based attentional decoder module. Details are as follows.

2.1. Data Preparation

For an input audio sample with a sampling rate of 44.1 kHz, 64 log mel-band energies are first extracted, using a Hamming window of 46 ms, with 50% overlap. We tokenize the captions of the development split with a one-hot encoding of the words. There are no unknown tokens/words since all the words in the development split appear in the other evaluation and test splits as well. $\langle SOS\rangle$ and $\langle EOS\rangle$ are also employed as the start-of-sequence and end-of-sequence tokens, respectively.

2.2. Down Sampling

The audio samples in Clotho have a uniform distribution between 15 and 30 s, which is not suitable for the network in both training and testing stages. Division schemes are often used in audio classification tasks to downsample the audio, however, an audio captioning system needs the whole audio to generate the description. In our system, average pooling function is applied to the log mel-band energies with a window size of 10 frames and a hop size of 5 frames. Thus, the network is easier to be trained and shows better performance.

2.3. CNN Encoder

In the baseline system, 3-layer bi-directional gated recurrent units (bi-GRUs) are applied as the encoder, which takes the log mel-band energies as input. However, the low-level features capture little time-frequency information of the audio and GRUs cannot model the frequency information. Inspired by the methods for the tasks of acoustic scene classification and sound event detection [6], convolutional neural network (CNN) is employed as the encoder in our system, which has powerful ability to model deep time-frequency representations. The details of the encoder architecture is shown in Table 1. To start with an input spectrogram of size $F\times T$, the convolutional layer consisting of C-channel filters outputs a $F'\times T'\times C$

[*] Yuexian Zou is the corresponding author.

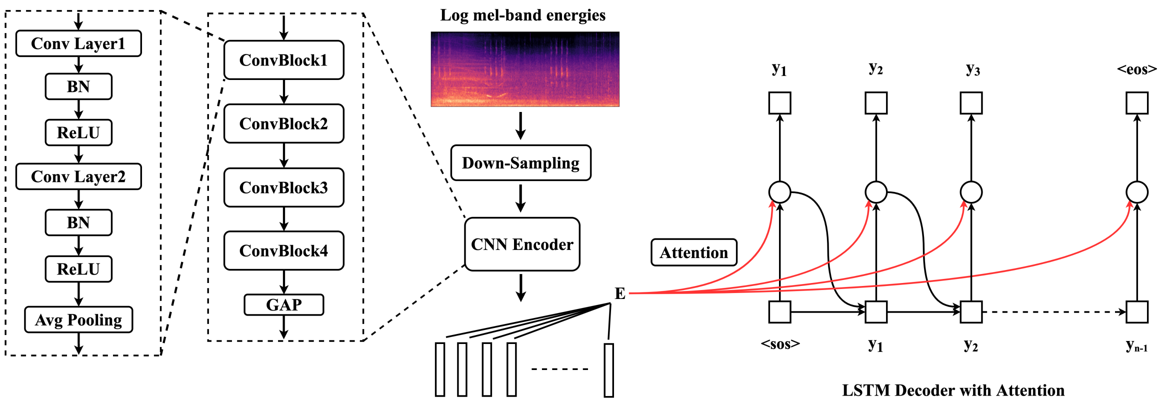

Figure 1: The illustration of our audio captioning system, which is composed of the down-sampling module, the CNN encoder module and the LSTM-based attentional decoder module.

Table 1: Encoder Architecture

Input: log mel-band energies $64 \times T$
Avg Pooling 1×10 (stride 1×5)
Conv 3×3 @ 64, BN, ReLU Conv 3×3 @ 64, BN, ReLU Avg Pooling 2×2 (stride 2×2)
Dropout 0.2
Conv 3×3 @ 128, BN, ReLU Conv 3×3 @ 128, BN, ReLU Avg Pooling 2×2 (stride 2×2)
Dropout 0.2
Conv 3×3 @ 256, BN, ReLU Conv 3×3 @ 256, BN, ReLU Avg Pooling 2×2 (stride 2×2)
Dropout 0.2
Conv 3×3 @ 512, BN, ReLU Conv 3×3 @ 512, BN, ReLU Avg Pooling 2×2 (stride 2×2)
Dropout 0.2
Avg Pooling 4×1 (stride 4×1)
Output: feature vectors $512 \times T/80$

feature map, which is then fed to the next convolutional layer to extract translation-shift invariant features. Our CNN is a VGG [7] style network, which consists of 4 convolutional blocks with the output channels of 64, 128, 256 and 512, respectively. Each convolutional block contains 2 convolutional layers with kernel size of 3×3, followed by downsampling with average pooling size of 2×2. Batch normalization and Rectified Linear Units (ReLU) are used following the convolutional operations. Global average pooling is applied to the frequency axis after the last convolutional block to obtain the final feature vectors. In addition, dropout with a ratio of 0.2 is applied between the convolutional blocks.

2.4. LSTM-based Attentional Decoder

The decoder consists of 1-layer LSTM and a classifier (a fully-connected layer), accepts the output of the encoder, and outputs a probability for each of the unique words. The decoder iterates for 22 time steps, which is the length of the longest caption.

Table 2: Comparison of metrics for the evaluation split of Clotho.

Metric	DCASE2020 Task6 Baseline	Ours
BLEU1	0.389	0.489
BLEU2	0.136	0.285
BLEU3	0.055	0.177
BLEU4	0.015	0.107
ROUGEL	0.262	0.325
METEOR	0.084	0.148
CIDEr	0.074	0.252
SPICE	0.033	0.091
SPIDEr	0.054	0.172

As conventional LSTM takes the hidden state of the previous time step as input and updates a new state, the information from the encoder may decrease step by step. In order to obtain the words of each time step with more acoustic information, temporal attention mechanism is applied in the decoder. To be specific, the hidden state of each time step depends on both the previous one and the encoder outputs. Here, we use the similar attention mechanism as the visual attention in [8]. In this case, the network can attend to salient part of the audio while generating its caption.

3. EXPERIMENTS

3.1. Experimental Setups

The decoder LSTM has 512 hidden units, and word embedding size is set to 512. To mitigate overfitting, we utilize dropout regularization with a rate of 0.5 in the decoder layers. The whole model is trained by Adam optimizer with an initial learning rate of 0.001. We stop training our model until 150 epochs are reached and the best model is selected according to the training loss. During the training process, scheduled sampling is applied with a linear decay from 1.0 to 0.7. While during the evaluation process, beam search with size 5 is used to generate sentences. In addition, data augmentation method SpecAugment [9] is applied in our experiments to prevent the system from over-fitting and improve the performance.

3.2. Experimental Results

Table 2 demonstrates the performance of our system and the baseline system on the evaluation split of Clotho dataset [2]. The results indicate that our system beats the baseline system under all the metrics.

4. CONCLUSION

In this technical report, we detailed our systems to tackle Task6 of the DCASE2020 challenge. Our system is a sequence-to-sequence model, including a encoder based on CNN and a LSTM-based attentional decoder. In future work, we would like to attempt other encoders or decoders for the audio captioning task.

5. ACKNOWLEDGMENT

This work was partially supported by Shenzhen Science & Technology Fundamental Research Programs (No: JCYJ20170817160058246 & JCYJ20180507182908274).

6. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 374–378.

[2] 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.

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

[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] http://dcase.community/challenge2020/.

[6] Q. Kong, Y. Cao, T. Iqbal, Y. Xu, W. Wang, and M. D. Plumbley, "Cross-task learning for audio tagging, sound event detection and spatial localization: Dcase 2019 baseline systems," *arXiv preprint arXiv:1904.03476*, 2019.

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[9] 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.