LEVERAGING STATE-OF-THE-ART ASR TECHNIQUES TO AUDIO CAPTIONING

Technical Report

Chaitanya Narisetty[1]*, Tomoki Hayashi*[2]*, Ryunosuke Ishizaki*[2]*,*
Shinji Watanabe[1]*, Kazuya Takeda*[2]

[1] Carnegie Mellon University, Pittsburgh, USA,
cnariset@andrew.cmu.edu, shinjiw@ieee.org
[2] Nagoya University, Nagoya, Japan,
{hayashi.tomoki, ishizaki.ryunosuke}@g.sp.m.is.nagoya-u.ac.jp,
takeda@is.nagoya-u.ac.jp

ABSTRACT

This report presents a summary of our submission to the 2021 DCASE challenge Task 6: Automated Audio Captioning. Our approach to this task is derived from state-of-the-art ASR techniques available in the ESPNet toolkit. Specifically, we train a convolution-augmented Transformer (Conformer) model to generate captions from input acoustic features in an end-to-end manner. In addition to the prescribed challenge dataset: Clotho-v2, we also augment the AudioCaps external dataset. To overcome the limited availability of training data, we further incorporate the Audioset-tags and audio-embeddings obtained by pretrained audio neural networks (PANNs) as an auxiliary input to our model. An ensemble of models trained over various architectures and input embeddings is selected as our final submission system. Experimental results indicate that our models achieve a SPIDEr score of 0.224 and 0.246 on the development-validation and development-evaluation sets respectively.

Index Terms— Automated Audio Captioning, Conformer, ESPNet, PANNs

1. INTRODUCTION

This paper describes the solution for DCASE 2021 Task 6: automated audio captioning task. The proposed method is based on state-of-the-art automatic speech recognition (ASR) techniques such as convolution-augmented Transformer (Conformer) architecture [1] and the fusion of a language model, which are incorporated with the end-to-end speech processing toolkit ESPnet [2]. Furthermore, we utilize the pretrained audio tagging model PANNs [3] to extract auxiliary information (e.g., Audioset [4] tags and embedding vector) and integrate them with the ASR model, enabling us to generate consistent captioning results. The contributions of this paper are summarized as follows:

- We apply an attention-based encoder-decoder with the Conformer architecture, which allows capturing both local and global contexts in the input sequence. We also employ the language model trained on the captions and integrate its score with shallow fusion, resulting in a more stable prediction [5].
- To further improve the performance, we introduce the pretrained audio tagging model PANNs to extract the auxiliary information, including Audioset tags and embedding vectors, and then utilize them as the additional inputs for the encoder-decoder model. The use of additional inputs leads to the generation of captions that have more variability.
- Experimental evaluation with DCASE 2021 Task 6 dataset [6] shows that the proposed encoder-decoder significantly outperforms the baseline and the use of additional information derived from the audio tagging model improves the performance. Our best model shows SPIDEr score of 0.224 and 0.246 on the development-validation and development-evaluation sets, respectively.
- Towards supporting accessible and reproducible research, we intend to release our audio captioning system and pre-trained models to the ESPNet toolkit[1] upon challenge completion.

2. PROPOSED METHODOLOGY

2.1. Overview

Fig. 1 illustrates an overview of the proposed method. Similar to other speech-related tasks, we use log-mel filterbank features as the primary input. In addition to the primary input, we employ auxiliary inputs such as Audioset tags and an embedding vector, which are extracted with the pretrained audio tagging model PANNs [3]. Both inputs are fed into the attention-based encoder-decoder model. Inspired by the success of Conformer-based models for tasks like speech recognition, translation, and separation [7], our model uses a Conformer encoder for processing these audio features and a Transformer decoder to process words in a corresponding caption. To further improve the performance, we introduce the RNN-based language model and combine it with the encoder-decoder model in the decoding stage. The following sections explain the details of each component and processing.

2.2. Conformer Encoder

The encoder incorporates a convolution sub-sampling layer and several Conformer blocks, where each block consists of a first feed-forward module, a multi-head self-attention module, a convolution module and a second feed-forward module in the aforementioned

[1] https://github.com/espnet/espnet

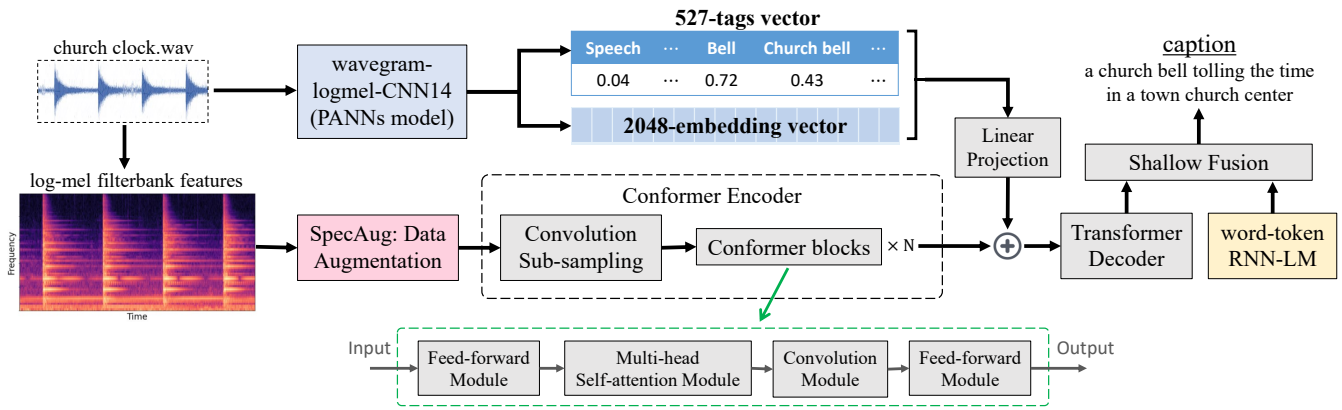

Figure 1: An overview of the proposed method based on a Conformer encoder and a Transformer decoder. SpecAug based data augmentation is performed on the log-mel filterbank features. The pre-trained wavegram-logmel-CNN14 PANNs model extracts the 527-tags vector and 2,048-embedding vector, and are fed as auxiliary inputs. Finally, a shallow fusion of decoder output and RNN-LM is performed to generate the output caption.

sequence. Similar to Transformer ASR models, a residual connection is added to the output of the feed-forward module followed by a layer normalization [8]. To regularize the network, the module further employs dropout and Swish activation [9].

Next, the self-attention module uses relative positional encoding in order to make the encoder robust to varying input length. This feature makes Conformer an ideal encoder for audio samples of varying length as seen in the present task. This module also employs dropout and a residual connection to regularize the network. Finally the convolution module employs a point-wise convolution, a gated linear unit (GLU) activation [10], 1-dim depth-wise convolution layer, a batch normalization layer [11], Swish activation and a point-wise convolution. A residual connection and dropout are again used for regularization.

2.3. Transformer Decoder

The decoder also incorporates several Transformer blocks, where each block consists of a multi-head self-attention layer, and a linear layer with ReLU activation sandwiched between two layer normalization layers.

2.4. Data Augmentation

We perform input data augmentation using SpecAug [12] consisting of three kinds of deformations - time warping, frequency masking and time masking. We set the maximum time warp parameter to $W = 5$, and randomly choose $w \in [0, W]$ such that the log-mel filterbank feature matrix is warped by w. Frequency and time masking are based on Cutout [13] regularization technique which masks a randomly chosen rectangular portion of the log-mel filterbank matrix. Dimensions of the mask were chosen randomly based on the maximum frequency and time masking parameters of $F_m = 30$ and $T_m = 40$ respectively.

2.5. Tags & Embeddings

To improve the generalization ability of our model, we provide an auxiliary input to our encoder framework, similar to the use of robust audio embeddings in speaker recognition tasks [14]. For this purpose, we use CNN14 - one the PANNs models trained on the large scale Audioset dataset of over 5,000 hours of audio samples labeled with 527 audio tags. The CNN14 model is a wavegram-logmel-CNN system trained on 32kHz audio samples using 14 convolution layers. The model outputs a 527-tags vector, whose each element corresponds to the prediction of an audio tag. In addition to this 527-tags vector, we also extract a 2,048-embedding vector from each audio sample that is inputted to final classification layer.

The tags and/or embeddings obtained using PANNs are used as an auxiliary input to our model. When using both the tags and embeddings, the two feature vectors are simply concatenated to form a single column vector. These features are first L2 normalized and then passed through a feed-forward layer to be projected to the same size as that of the attention dimension. The projected features are finally added to the output of the Conformer encoder, before being sent as an input to the Transformer decoder.

2.6. Ensemble Decoding and Shallow Fusion with LM

For better predictive performance, we implement an ensemble decoding module which performs posterior averaging of the attention score output from the several model decoders. We also separately train an word-token RNN language model (RNN-LM) using the captions in the training data. During inference, we integrate the decoder and separately trained RNN-LM with shallow fusion [5].

3. EXPERIMENTS

3.1. Data Preparation

Our proposed model takes 16 kHz audio samples as input and computes 80 log-mel energies from each 64 ms frame, shifted every 32 ms. Accordingly, all the audio files in Clotho-v2 dataset were downsampled from 44.1 kHz to 16 kHz. The overall development split of the Clotho-v2 dataset has 3,839 training samples, 1,045 validation samples and 1,045 evaluation samples. Each audio sample is 15-30 seconds long and contains 5 human generated captions with 8-20 words each.

Since the challenge dataset is relatively small to train large neural networks, we additionally augment the training data with roughly 46,000 single caption audio samples from the AudioCaps dataset [15]. Audio samples in this dataset are carefully chosen from the 2M samples in Audioset dataset [4]. Each audio sample is roughly 10 seconds long.

3.2. Model Variations

The baseline Conformer model used in our experiments has 16 encoder layers and 4 decoder layers, each with 1,024 units along with 256-dim attention layers with 4 heads and a depth-wise convolution with kernel size of 15. A variation of the baseline Conformer model was trained with smaller encoder-decoder layers having 512 units each. Another variation was trained with a smaller attention framework having 128-dim layers with 2 heads. Final model variation was trained with above mentioned smaller attention framework, but with a larger kernel size of 31.

3.3. Additional Input Features

In addition to the log-mel energies, we extract a softmax vector of 527-tags and a 2,048-embedding vector from each audio sample using the CNN14 PANNs model [3]. Each element in the 527-tags vector represents the probability of a corresponding class-label in the Audioset ontology. Another variation of the baseline Conformer model was trained with these extracted 2,048-embeddings and/or 527-tags as additional inputs to the encoder-decoder model.

3.4. Hyper-Parameters

During training, 64 audio-caption pairs were batched together and trained for 50 epochs with a learning-rate of 0.5, dropout of 0.1, cross-entropy loss function and *noam* optimizer [16]. To prevent exploding gradients, we set the gradient threshold to 5. Label smoothing [17] was set to 0.1 to avoid high confidence training predictions.

Upon completion of training, we average the model parameters over the final-10 epochs and this averaged model was used for inference. During inference, beam search was performed with a beam-size of 10 and RNN-based language model weight of 0.2. We note that the above hyper-parameters are optimized based on our prior experience in tuning ASR systems.

3.5. Results

The performance of our trained models were evaluated on both the development-validation and development-evaluation splits and are summarized in Table 1 and Table 2. We observe a slight degradation in performance when varying our model's architecture as compared to the baseline Conformer. However these variations help to improve the performance of a model ensemble. Secondary input features of tags and embeddings were able to improve the performance, especially over the development-validation split.

We also observe that augmenting the training data with the development-evaluation split was indeed able to improve the baseline Conformer's performance over the development-validation split and vice-versa. Finally, model ensembling was performed over various combinations of our trained models and the best performing ensembles were chosen for our final submission.

4. CONCLUSION

The present technical report details our submission to the DCASE 2021 Challenge Task 6: automated audio captioning. The proposed methodology employs existing state-of-the-art ASR techniques including Conformer-encoder, Transformer-decoder, data augmentation, embeddings as auxiliary inputs and shallow fusion with a pretrained RNN language model. Our experiments qualify the ability of these techniques for effective captioning of audio samples.

5. ACKNOWLEDGMENT

This work was supported in part by Sony Corporation, JHU HLTCOE, and Bridges PSC (TG-CIS210014).

6. REFERENCES

[1] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[2] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[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. [Online]. Available: https://doi.org/10.5281/zenodo.3987831

[4] 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.

[5] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Proceeding of Interspeech*, 2017, pp. 949–953. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-1296

[6] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020. [Online]. Available: https://arxiv.org/abs/1910.09387

[7] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[8] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[9] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

Method	BLEU-1,2,3,4				ROUGE-L	METEOR	CIDEr	SPICE	SPIDEr
Conformer	0.512	0.317	0.205	0.131	0.336	0.148	0.310	0.100	0.205
smaller enc-dec	0.500	0.311	0.203	0.129	0.336	0.144	0.299	0.099	0.199
smaller attention	0.490	0.307	0.199	0.127	0.332	0.143	0.310	0.096	0.203
+ larger-kernel	0.496	0.307	0.198	0.124	0.336	0.143	0.297	0.098	0.198
+ 2048-embeddings	**0.527**	0.329	0.214	0.136	**0.346**	**0.154**	**0.325**	0.102	0.214
+ 527-tags	0.513	0.321	0.208	0.130	0.337	0.150	0.315	0.098	0.207
++ 2048-embeddings	0.521	**0.330**	**0.217**	**0.138**	0.345	**0.154**	0.323	**0.107**	**0.215**
+ dev-eval split	0.515	0.321	0.207	0.131	0.340	0.149	0.314	0.101	0.208
Best Ensemble	0.533	0.343	0.226	0.146	0.355	0.154	0.341	0.106	0.224

Table 1: Scores of evaluation metrics for the development-validation split.

Method	BLEU-1,2,3,4				ROUGE-L	METEOR	CIDEr	SPICE	SPIDEr
Conformer	0.534	0.343	**0.233**	**0.158**	0.354	0.157	0.351	0.106	0.228
smaller enc-dec	0.524	0.331	0.219	0.144	0.356	0.153	0.329	0.103	0.216
smaller attention	0.506	0.320	0.212	0.140	0.349	0.152	0.337	0.102	0.219
+ larger-kernel	0.518	0.330	0.224	0.150	0.355	0.154	0.340	0.105	0.223
+ 2048-embeddings	0.533	0.338	0.222	0.145	0.353	0.157	0.346	0.104	0.225
+ 527-tags	0.534	0.339	0.223	0.144	0.355	0.159	0.342	0.106	0.224
++ 2048-embeddings	0.536	0.341	0.225	0.146	0.357	0.160	0.346	0.108	0.227
+ dev-val split	**0.541**	**0.346**	0.231	0.152	**0.356**	**0.161**	**0.362**	**0.110**	**0.236**
Best Ensemble	0.546	0.356	0.243	0.165	0.369	0.163	0.381	0.110	0.246

Table 2: Scores of evaluation metrics for the development-evaluation split.

[10] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[12] 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.

[13] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[15] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *NAACL-HLT*, 2019.

[16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.