CAU SUBMISSION TO DCASE 2021 TASK6: TRANSFORMER FOLLOWED BY TRANSFER LEARNING FOR AUDIO CAPTIONING

Technical Report

Hyejin Won[1], *Baekseung Kim*[1], *Il-Youp Kwak*[1], *Changwon Lim*[1]

[1]Chung-Ang University, Department of Applied Statistics, Seoul, South Korea,
{whj9492, kbs778, ikwak2, clim}@cau.ac.kr

ABSTRACT

This report proposes an automated audio captioning model for the 2021 DCASE audio captioning challenge. In this challenge, a model is required to generate natural language descriptions of a given audio signal. We use pre-trained models trained using AudioSet data, a large-scale dataset of manually annotated audio events. The large amount of audio events data would help capturing important audio feature representation. To make use of the learned feature from AudioSet data, we explored several transfer learning approaches. Our proposed sequence-to-sequence model consists of a CNN14 or ResNet54 encoder and a Transformer decoder. Experiments show that the proposed model can achieve a SPIDEr score of 0.246 and 0.285 on audio captioning performance.

Index Terms— audio captioning, acoustic event detection, transfer learning, deep learning

1. INTRODUCTION

This Technical Report was written to describe the model of Automated Audio Capturing (AAC) addressed from task 6 of the DCASE 2021 challenge [1]. The goal of the model is to automatically generate captions on a given sound data. One example of generated caption on a given sound could be "people **talking** in a **small** and **empty room**". The 2021 DCASE AAC uses the Clotho v2.1 dataset [2], which has more data than the Clotho v1 dataset. Clotho v2.1 dataset contains 6,974 (4,981 from version 1 and 1,993 from version 2.1) audio clips in 15-30 seconds each with 5 captions in 8-20 English words. This year, the use of external data is allowed. The important sound related feature representation could be trained using a massive amount of data such as AudioSet which include 2,084,320 human-labeled 10-second sound clips on 632 audio event classes. With the transfer learning, our proposed model took two pre-trained networks, 14-layers CNN (CNN14) and 54-layers ResNet (ResNet54), trained on AudioSet as the encoder part [3, 4]. We trained a transformer decoder using the Clotho v2.1 dataset for natural language generation.

2. PROPOSED MODEL

2.1. System Overview

The Figure 1 shows the proposed system overview. The pre-trained CNN14 and ResNet54 are taken as an encoder of our proposed model using transfer learning [4]. We used a transformer decoder and trained our model using the Cloth v2.1 dataset.

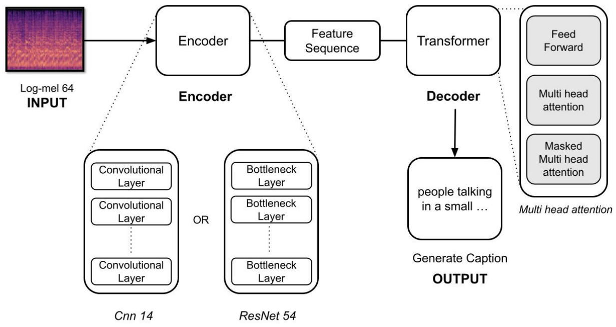

Figure 1: Model Architecture

2.2. Pre-Processing

The log-mel spectrogram feature is used for the input feature. Audio data have 44.1kHz sampling frequency, and we applied Hann window of 1024 size with 50% overlaps. From each window frame we extracted 64 log mel-band energies. For the number of time windows, we calculate the maximum time window number, T, among sample datasets. We zero padded the time dimension to size T for the fixed size input feature on our model.

The word embedding is pre-trained using Word2Vec model [6] via python package gensim [7]. Each caption sentence in the training set is used to form a training corpus.

2.3. Data Augmentation

Spec Augment [8] is applied as a data augmentation method for more robust training. With the Spec Augment, frequency masks and time masks are randomly applied onto the log-mel spectrogram before we feed the log-mel spectrogram input to the CNN14 or ResNet54 encoder.

2.4. Pretrained Audio Neural Networks using AudioSet

Kong et al. (2020) proposed Pretrained Audio Neural Networks(PANNs) trained on the large-scale AudioSet dataset, and made the pretrained models, CNN14 and ResNet54, available to

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (2020R1C1C1A01013020, 2021R1F1A1056516)

Table 1: CNN14 architecture

CNN14
Log-mel spectrogram 64 mel bins
(3 × 3 @64,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @128,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @256,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @512,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @1024,BN,ReLU)×2
Pooling 2 × 2
(3 × 3 @2048,BN,ReLU)∗2

Table 2: ResNet54 architecture

ResNet54
Log-mel spectrogram 64 mel bins
(3 × 3 @512,BN,ReLU)×2
Pooling 2 × 2
(bottleneckB@64)×3
Pooling 2 × 2
(bottleneckB@128)×4
Pooling 2 × 2
(bottleneckB@256)×6
Pooling 2 × 2
(bottleneckB@512)×3
Pooling 2 × 2
(3 × 3 @512,BN,ReLU)×2

the public. The AudioSet dataset contains over 5,000 hours of audio recording with 527 sound classes. The pre-trained model is a multi-label classification model for 527 sound classes. The same log-mel spectrogram feature in the Pre-Processing section is used for the input data for the classification model. These PANNs could be transferred to other audio related tasks. We took CNN14 and ResNet54 as the encoder part for the AAC model. The Table 1 and 2 describe CNN14 and ResNet54 model architecture, respectively.

2.5. Proposed Model

Our model uses CNN14 or ResNet54 as an encoder for feature extension and Transformer Decoder for natural language generation. We take pre-trained networks (CNN14, ResNet54) learned from PANNs to AudioSet and use them as our encoder. We freeze the weights learned from pre-trained networks and bring them to our model. Furthermore, we attempt fine-tune encoder network by unfreezing last convolution block layers of CNN14 and ResNet54 to find the optimal model.

2.5.1. Encoder

Our model uses Resnet54 and CNN14 as an encoder for feature extraction of input log-mel spectrogram [4]. Table 1 and 2 show the structure of the CNN14 and ResNet54 that we used for ACC, respectively the number after the "@" symbol indicates the number of feature maps. BottleneckB is abbreviation for bottleneck block.

Table 3: SPIDEr score for model performance on evaluation data

Model	**SPIDEr Score**
Baseline Model	0.054
CNN14 + Transformer (From Scratch)	0.148
ResNet54 + Transformer (From Scratch)	0.133
CNN14 + Transformer (Transfer-learning with finetuning)	0.171
ResNet54 + Transformer (Transfer-learning with finetuning)	0.159
CNN14 + Transformer (Transfer-learning)	0.285
ResNet54 + Transformer (Transfer-learning)	0.246

2.5.2. Decoder

It uses a standard transformer decoder consisting of multi-head self-attention as a decoder. The decoder uses a 2-layers transformer with a hidden dimension of 192 and 4 heads. Transformer model as the decoder helps prevent gradient vanishing or exploding.

3. EXPERIMENTS

3.1. Experimental Setups

In training, batch size of 8 is used with a learning rate of 10^{-4} and a l2 regularization applied to all trainable parameters with factor λ = 10^{-6}. We use the Adam Optimizer [9] and apply the Stochastic Weight Averaging (SWA) method [10] to boost performance. Dropout in $P = 0.2$ is applied to ResNet54 encoder and Transformer decoder. In the training process, each audio is combined with each one of five caption annotations and used as a sample. In the evaluation, each audio is used as one sample and all five captions are used as reference for metric computation. The log-mel spectrogram input is obtained by first getting the 64 Mel-band log-mel spectrogram of the audio, then converting the amplitude into a decibel scale. In the inference stage, a beam search with a beam size of 3 is implemented to achieve better decoding performance. The Word2Vec model is trained 1000 epochs with random parameter initialization. The proposed model is trained 30 epochs before the model with the highest performance is selected for fine-tuning. The selection of the model is based on SPIDEr score [11] of the evaluation performance. As the challenge allows to submit up to 4 results, 4 models that has the highest SPIDEr score is selected for result submission.

3.2. Experimental Results

Table 2 shows that the CNN14 Encoder+Transformer Decoder and ResNet54 Encoder+Transformer Decoder model have a higher SPIDEr score than the baseline model. This shows that the transfered encoder trained with sufficiently large amount of audio data performs well on AAC. We also experimented fine tuning last convolution block of encoder networks (CNN14 and ResNet54). However, fine tuning did not worked well for both encoders of CNN14 and ResNet54.

4. CONCLUSION

The pre-trained networks using a large amount of audio data would capture important audio features. Since the competition this year allows the use of other datasets, we transferred CNN14 and ResNet54 network trained on AudioSet data as the encoder part of the proposed system to make use of the valuable pre-trained network. With the transferred network and a transformer decoder our system worked well with SPIDEr score of 0.285 for CNN14 encoder and 0.246 for ResNet54 encoder. Further, we experimented training all the networks from scratch, transfer learning with fine tuning, and transfer learning without fine tuning. Among them, transfer learning without fine tuning worked the best.

5. REFERENCES

[1] Drossos, Konstantinos, Sharath Adavanne, and Tuomas Virtanen. ”Automated audio captioning with recurrent neural networks.” *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA). IEEE, (2017).*

[2] Drossos, Konstantinos, Samuel Lipping, and Tuomas Virtanen. ”Clotho: An audio captioning dataset.” *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, (2020).*

[3] He, Kaiming, et al. ”Deep residual learning for image recognition.” *Proceedings of the IEEE conference on computer vision and pattern recognition. (2016).*

[4] Kong, Qiuqiang, et al. ”Panns: Large-scale pretrained audio neural networks for audio pattern recognition.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020): 2880-2894.*

[5] Vaswani, Ashish, et al. ”Attention is all you need.” *arXiv preprint arXiv:1706.03762 (2017).*

[6] Mikolov, Tomas, et al. ”Efficient estimation of word representations in vector space.” *arXiv preprint arXiv:1301.3781 (2013).*

[7] Rehurek, Radim, and Petr Sojka. ”Software framework for topic modelling with large corpora.” *In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks. Valletta, Malta: ELRA, May (2010), pp. 45-50.*

[8] Park, Daniel S., et al. ”Specaugment: A simple data augmentation method for automatic speech recognition.” *arXiv preprint arXiv:1904.08779 (2019).*

[9] Kingma, Diederik P., and Jimmy Ba. ”Adam: A method for stochastic optimization.” *arXiv preprint arXiv:1412.6980 (2014).*

[10] Izmailov, Pavel, et al. ”Averaging weights leads to wider optima and better generalization.” *arXiv preprint arXiv:1803.05407 (2018).*

[11] Liu, Siqi, et al. ”Improved image captioning via policy gradient optimization of spider.” *Proceedings of the IEEE international conference on computer vision. (2017).*