THE DCASE2021 CHALLENGE TASK 6 SYSTEM : AUTOMATED AUDIO CAPTION

Technical Report

Liu Yang

Beijing Institute of Technology
3120200800@bit.edu.cn

Bi Sijun

Beijing Institute of Technology
3120200743@bit.edu.cn

ABSTRACT

This technical report describes the system participating to the Automated Audio Captioning (DCASE) 2021 Challenge, Task 6: automated audio captioning.In this work, We employ several learnable stack CNNs to extract audio features in the encoder layer, meanwhile, we employ the decoder of the widely used Transformer structure to generate captions. For optimize system, we use the sentence-level cosine loss function and crossentropy loss. The experimental results show that our system could achieve the SPIDEr of 0.166 on the evaluation split of the Clotho dataset.

Index Terms— AAC, CNNs, Transformer, Sentence-level cosine loss function

1. INTRODUCTION

Sound signal is one of the sources that people receive the most information from the outside world. The automated audio captioning problem refers to the task of describing an audio signal in text. This technical report describes the system participating to the Detection and Classification of Acoustic Scenes and Events(DCASE), 2021 Challenge, Task 6: automated audio captioning. It can visualize sound information, and through its description of sound signal, we can receive sound information in other ways. The automatic audio captioning system for this task is not a conventional voice-to-text system, it is an inter modal translation task, where a system accepts an audio signal as an input and outputs the textual description (i.e. the caption) of that signal. AAC methods can model concepts (e.g. "heavy rainfall"), physical properties of objects and environment (e.g. "A shower hisses as water flows"), and high level knowledge ("The church bell chimes three times and echoes loudly"). This modeling can be used in various applications, ranging from automatic content description to intelligent and content oriented machine-to-machine interaction.

In this report we use the widely used seq2seq structure to generate the captions of audio, in the encoder layer, we use a learnable stack CNNs to extract features which includes the information need for AAC. As for the CNNs, We adopt existing machine listening approaches where sound sources and actions are well captured by time-frequency information [1,2]. In the decoder layer, we use the decoder of the Transformer[3] to decoder the audio features and generated captions. For optimize system, we use the sentence-level cosine loss function and crossentropy loss. The sentence-level cosine loss can help generating the right order captions while the crossentropy loss

can help recall the true words. Finally, We assess our method utilizing the freely available splits of Clotho datasets[4].

2. SYSTEM DESCRIPTION

Our proposed method takes an audio file with 44.1kHz sampling rates, At first, a matrix of features, $\mathbf{X} \in \mathrm{R}^{\mathrm{Ta} \times \mathrm{F}}$, are extracted from audio files as our system input, which Ta means the time dimension, F means the log-mel dimension, the initial audio features is extracted by the way same as the dcase2021 baseline system. And the outputs of our system is a sequence vector, $\mathbf{Y} \in \mathrm{R}^{\mathrm{L} \times \mathrm{W}}$, which L represents the sentence length, W represents the the size of the vocabulary, we use the beam search method to get the final captions, our method utilizes an encoder-decoder scheme, where the encoder is based on stack CNNs and the decoder is based on feed-forward neural networks (FFNs) and multi-head attention which same as the decoder layer of the transformer. We mainly follow [5] to build our model.
It's result is then output to the linear layer. In the training process, sentence-level of loss function and crossentropy loss is used for optimize system.

2.1. Encoder

Through the time-frequency analysis, each signal is transformed into a map in the time domain and frequency domain, which fully explains the energy change of the signal in time and frequency. Meanwhile, since the convolutional neural network (CNN) is an excellent model for feature extraction in computer vision, this paper regards such signal graph as the input of CNN to extract features.

To extract audio features by CNN, we first stack the CNN layers and get the stack CNN block, then we get the encoder layer by reuse the stack CNN block Ne times.

The structure of the stack CNN block is shown on the left of the figure 1, the first input of the stack CNN block is the audio features while others is the last output of the stack CNN block. In the stack CNN block, The input vector is first pass the depthwise separable convolution $\mathrm{CNN1}^{\mathrm{Ne}}$ which Ne means the Ne CNN block, and then pass the Leaky Relu activation layer and the Batch normalization(BN) layer, then the output is pass a ordinary 2D convolution neural network $\mathrm{CNN2}^{\mathrm{Ne}}$ instead of the depthwise separable convolution, $\mathrm{CNN2}^{\mathrm{Ne}}$ layer is followed by a Relu layer, a BN layer, a max pooling layer and a Dropout layer.

CNN1^{Ne} has $C_{in}^{ne} = 1$, $C_{out}^{ne} = C_{out}^{ne}$ input and output channels, respectively, and it's kernel size is (5, 5) , with unit stride, and padding of 2. CNN2^{Ne} consists of a square kernel of size $K_{CNN2} > 1$, with unit stride, and padding of 2, the input channels and the output channels of CNN2^{Ne} are the C_{out}^{ne} . The final output of the encoder layer will be reshaped into Z = {1 × Ta × C_{out}^{ne} }

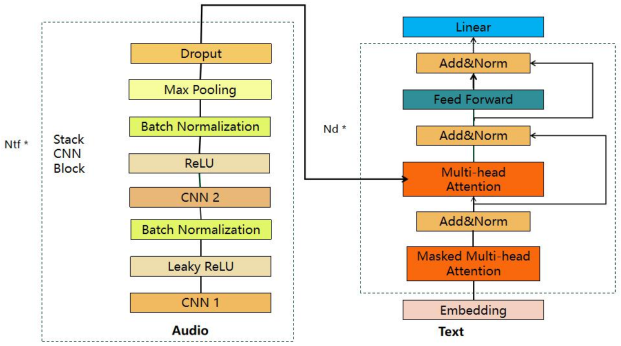

Figure 1: The system, with the encoder on the left-hand side and the decoder on the right-hand side

2.2. Decoder

We employ the decoder of the Transformer model [3] as our decoder, the basic structure is as same as [3], we use Nd atteneion block in the decoder and Nh Head, the structure is shown in figure, the input of the decoder is formed by the output of the encoder and the previously generated words. The final output of the decoder is $Y \in R^{L \times W}$. When generate the captions, we use the beam search method instead of the greedy search.

2.3. Sentence-level loss

In addition to the standard CE loss at word-level, we use a sentence-level loss to capture semantic similarity better. Since the input of the final Linear layer of the decoder can be seen as the predicted word embedding at each time step, we first pool it to get a single representation of the prediction, which can be seen as the predicted sentence loss. As Equation (1) shows, we use mean pooling on all ht to obtain the representation e^s.

$$e_{\hat{s}}(\theta,F)=\frac{1}{T}\sum_{t=1}^{T}h_{t}(\theta,F) \qquad (1)$$

In order to minimize the embedding difference between predicted sentence e^s and annotated sentences es (which get by the word2vec model trained on the ground truth captions), we use a sentence loss function opposed to cosine similarity, Equation (2). In this way, a small sentence loss indicates a high semantic similarity.

$$1_{setence}(\theta;e_{s};F)=1-\frac{e_{s}\cdot e_{\hat{s}}(\theta,F)}{\max\left(\left\|e_{s}\right\|_{2}\cdot\left\|e_{\hat{s}}(\theta,F)\right\|_{2}\right)} \qquad (2)$$

The whole loss is the crossentropy loss added with weighted sentences loss, which weight is set to 10 in our model.

2.4. Training set

The hyper-parameters of the audio pre-processing are as follows. All audio samples were down-sampled at 44.1 kHz. The number of mel-filterbank was F = 64. we use Ne = 3, C_{out}^{ne} =128， Nd =3, dropout=0.25, Nh = 4, beam_size=2, embedding_size = 128. while training, we use Adam optimizer and clipping of the 2-norm of the gradients to the value of 1. we use the split DCASE Challenge datasets to train the model.

3. RESULTS

We evaluate our model on the development evaluation split of the Clotho datasets, the result is shown in the table 1.

BLEU1	0.483
BLEU2	0.298
BLEU3	0.197
BLEU4	0.119
METEOR	0.133
ROUGE_L	0.322
CIDER	0.243
SPICE	0.088
SPIDER	0.166

4. REFERENCES

[1] K. Drossos, S. I. Mimilakis, S. Gharib, Y. Li, and T. Virtanen, "Sound event detection with depthwise separable and dilated convolutions," in 2020 International Joint Conference on Neural Networks (IJCNN), 2020.

[2] E. C¸ akır, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional recurrent neural networks for polyphonic sound event detection," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 6, pp. 1291–1303, 2017.

[3] A. Vaswani et al., "Attention is all you need," in 31st Conference on Neural Information Processing Systems (NeurIPS 2017), 2017.

[4] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

[5] A. den Oord et al., "Wavenet: A generative model for raw audio," in 9th International Speech Communication Association (ISCA) Speech Synthesis Workshop, 2016.