AUTOMATED AUDIO CAPTIONING WITH MLP-MIXER AND PRE-TRAINED ENCODER

Technical Report

Feiyang Xiao[1], *Jian Guan*[1*], *Qiuqiang Kong*[2]

[1] Group of Intelligent Signal Processing, College of Computer Science and Technology
Harbin Engineering University, Harbin, China
{xiaofeiyang128@gmail.com, j.guan@hrbeu.edu.cn}
[2] ByteDance, Shanghai, China
{kongqiuqiang@bytedance.com}

ABSTRACT

This technical report describes the submission from the Group of Intelligent Signal Processing (GISP) for Task6 of DCASE2021 challenge (automated audio captioning). Our audio captioning system is based on the sequence-to-sequence autoencoder model. Previous recurrent neural network (RNN) and Transformer based methods just perceive the time dimension information but ignore the frequency information. To utilize both time dimension and frequency dimension information, multi-layer perceptrons mixer (MLP-Mixer) is used as the encoder. For caption prediction, a Transformer decoder structure is used as the decoder. No extra data is employed. In addition, to highlight the content information, we use a pre-trained encoder with multi-label content information. The experimental results show that our system can achieve the SPIDEr of 0.144 (official baseline: 0.051) on the evaluation split of the Clotho dataset. In addition, comparing with Transformer methods, our system has fewer training time.

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

1. INTRODUCTION

The automated audio captioning (AAC) is an intermodal translation task which translates an input audio into a corresponding description (i.e., caption) by using natural language methods [1–3]. This task expects that the caption is as close as possible to an unartificial one. AAC is different from the sound event detection (SED) and the acoustic scene classification (ASC) tasks. AAC does not predict a sound event/scene, but describes the general information including the identification of sound events, acoustic scenes, foreground versus background discrimination, concepts and physical properties of objects and environments [3]. AAC has positive effects in various applications, such as intelligent and content oriented machine-to-machine interaction and automatic content description [4].

Apparently, AAC seems like automated image captioning which describes information from an input image [5, 6]. However, there are some significant differences [7]. First, the audio can get the information that is unable to provide in an image, because the sound includes multi-directional information. Second, the spectrogram of the audio includes both time and frequency features, which is different from the space feature in images [8]. Third, audio signal is time series signal with natural temporal information, but the image does not [1]. The research of automated audio captioning will get information which cannot be perceived by the optical image.

This report describes the details of the GISP team's submission for Task6 of DCASE2021. Our system is a sequence-to-sequence autoencoder model, which contains an encoder based on Multi-layer perceptrons mixer [9] and a self-attention decoder based on Transformer [10]. A pre-trained encoder with multi-label content words is also employed. The experimental results shows the effectiveness of our AAC system. On the official evaluation split of Clotho, our system can achieve the SPIDEr of 0.144.

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

2. SYSTEM STRUCTURE

In this section, we give the architecture of our AAC system, as shown in Figure 1. Specifically, it is based on a sequence-to-sequence autoencoder model, including an audio embedding module, an MLP-Mixer encoder module and a self-attention Transformer decoder module. The details are given as follows.

2.1. Audio Embedding

The input audio feature of our system is the log-mel spectrogram. Note that, input log-mel spectrograms have diverse time frames and have the same frequency feature dimension (i.e., 64). We use a 1D-convolutional neural network (CNN) to upsample the frequency dimension from 64 to 128, getting the audio embedding from the log-mel feature for the MLP-Mixer encoder. The details of the audio embedding is provided in Table 1.

Table 1: Audio embedding process.

Input: log-mel spectrogram $(B \times T \times F)$
Rearrange $(B \times T \times F)$ to $(B \times F \times T)$
1D-CNN 1×10 @ 128, (stride 1×5)
Rearrange $(B \times H \times T')$ to $(B \times T' \times H)$
Output: audio embedding feature $(B \times T' \times H)$

[*] Jian Guan is the corresponding author.

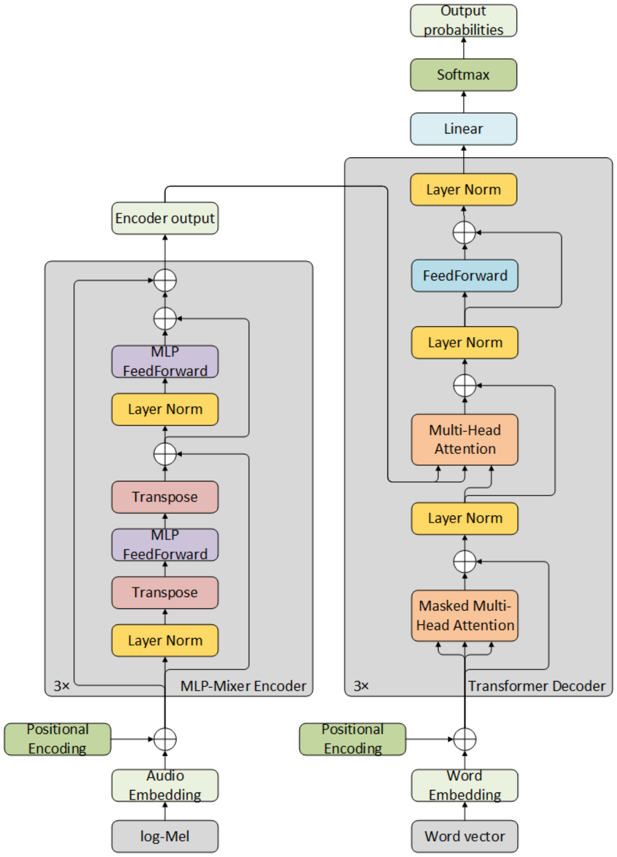

Figure 1: The proposed automated audio captioning system.

We set the size of a batch input spectrograms as $B \times T \times F$. B represents the batch size, T represents the time dimension, and F represents the frequency dimension. Then the input feature will be transposed to the size of $B \times F \times T$, and passed to a 1D-CNN module. This 1D-CNN module can get F channels features and output the audio embedding features with H filters. Here, H represents the embedding size (i.e., 128). Note that, we set the 1D-CNN with the kernel size of 1×10 and stride of 1×5, to squeeze the dimension of time sequence and extract the efficient feature. Finally, after a transposing process, the size of the audio embedding feature is $B \times T' \times H$. Here, T' represents the time dimension after 1D-CNN.

2.2. MLP-mixer Encoder

In the baseline system of the Task6, 3-layer bi-directional gated recurrent units (bi-GRUs) are applied as the encoder, and the input data feature of the baseline is the log-mel spectrogram. The bi-GRUs encoder can model the time series information of the log-mel spectrogram, however, it cannot perceive the frequency information in the input feature. In order to model the time domain and the frequency domain simultaneously, we employ multi-layer perceptrons mixer (MLP-Mixer) structure as the encoder of our system, due to its ability to perceive information in both time domain and frequency domain.

As shown in Figure 1, the MLP-Mixer encoder has three MLP-Mixer encoder layers. Each encoder layer can be seen as an MLP-Mixer block, which employs transposing and MLP feedforward module. Transposing process exchanges the dimensions of the time domain and the frequency domain. Because of the transposing process, MLP-Mixer blocks can perceive the time domain and the frequency domain simultaneously with MLP feedforward modules. The layer norm in our model is used to normalize the feature, and the skip connection is used to avoid the gradient vanishing. Meanwhile, we also use the skip connections of the input and each encoder layer output, to obtain the features of different level via summation operation. Here, the MLP feedforward module consists of two linear layers and activated by a GELU function layer, as shown in Figure 2.

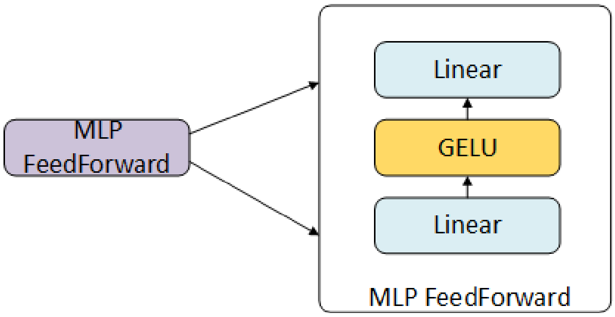

Figure 2: The structure of the MLP feedforward module.

2.3. Transformer Decoder

The decoder of our system is the same as that of the Transformer model in [10]. The decoder in our system consists of three transformer decoder layers.

The input data of our decoder is the word embedding feature. For a decoder layer, the input is passed to the masked multi-head attention module and output a query vector feature. Then the value vector becomes one of the inputs for the next multi-head attention module. The multi-head attention module also uses the output of the MLP-Mixer encoder as the key vector input and the value vector input. After layer norm and feedforward module, we can obtain the output of a decoder layer. In each decoder layer, skip connection strategy is also used to avoid gradient vanishing problem.

Finally, the output of the decoder is passed to a linear layer and a softmax function to get the output probabilities of the caption words. Note that, during the training stage, our system uses teacher forcing strategy to train the decoder, which uses the original captions to word embedding as the input data of the decoder module.

During the evaluation stage, we choose beam search strategy [11] to predict the caption of the audio signal. Beam search strategy can preserve the top k possible words for each word prediction process with previous prediction results, and output the most possible caption finally. In our system, k is empirically set as 5.

2.4. Pre-trained encoder

The MLP-Mixer encoder is used to model the content of audio which will be passed to the Transformer decoder to predict caption words. However, there are more function words (e.g., a, an, the, and etc) in captions without any essential meaning than the content words (i.e., adjectives, nouns), which may let the system identify

Table 2: Performance comparison on Clotho dataset.

Metric	Baseline [3]	Transformer	MLP/w/AE	MLP/w/Pre	Proposed system
BELU$_1$	0.378	0.450	0.460	0.471	0.461
BELU$_2$	0.119	0.262	0.266	0.282	0.275
BELU$_3$	0.050	0.167	0.170	0.182	0.180
BELU$_4$	0.017	0.105	0.104	0.112	0.112
ROUGE$_L$	0.263	0.306	0.307	0.317	0.312
METEOR	0.078	0.124	0.124	0.128	0.126
CIDEr	0.075	0.206	0.198	0.208	0.210
SPICE	0.028	0.071	0.073	0.078	0.079
SPIDEr	0.051	0.138	0.136	0.143	**0.144**

the content of audio obscurely. To mitigate this problem, we adopt a pre-trained encoder strategy. The process of this strategy is shown in Figure 3. According to [12], first, we abandon 20 words with highest frequency and the words whose length are less than 3 letters. Then, we convert the words with ’-ing’, ’-ly’, ’-d’, ’-s’, etc., to their original words and add their frequency. Finally, we choose 300 words with the highest frequency as the multi-label for pre-trained encoder. Note that, all 5 captions of each audio has the same multi-label, such that an audio just needs to be trained once in an epoch. During the captioning prediction training, the pre-trained encoder module will be loaded as the MLP-Mixer encoder as shown in Figure 1, and optimized by the loss function of captioning task.

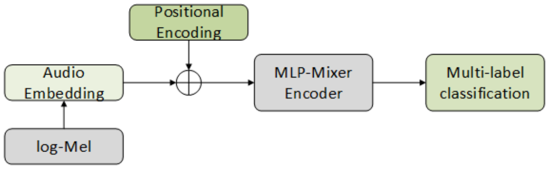

Figure 3: The process of the pre-trained encoder.

3. EXPERIMENTS AND RESULTS

3.1. Data Pre-processing

Our system works on the Clotho (v2) dataset from Task6 of DCASE2021. Clotho dataset consists of audio samples of 15 to 30 seconds duration, with each audio sample having five captions of 8 to 20 words length. There is a total number of 6,974 audio samples in Clotho, with 34,870 captions. The dataset is divided into four splits: development, validation, evaluation, and testing. Our system is trained by development set and validation set, and evaluated on evaluation set. Finally, we submit the evaluation results on test set.

Experiments use log-mel spectrograms as audio input feature, which comes from the raw audio signals with a sample rate of 44.1 kHz. We get 64 log-mel band spectral, using Hamming window with 50% overlap. For the unity of encoder dimensions, we pad the audio spectral to the max time sequence length T (i.e., $T = 2,584$) with 0.

We tokenize the captions of the development set. There are no unknown tokens/words since all the words in the development set appear in the validation set, evaluation set and test set. $<$ sos $>$, $<$eos$>$ and $<$pad$>$ are employed to denote the start-of-sequence, the end-of-sequence and sequence padding, respectively. In a batch word vectors input, we pad the word vectors to the max length of this batch with $<$pad$>$, and the max length of the whole dataset is 22.

3.2. Experimental Setup

The time sequence dimension and the frequency dimension of the MLP-Mixer encoder are 515 and 128, respectively. The hidden dimension of time sequence processing is 128 and the hidden dimension of frequency processing is 64. The model dimension of Transformer decoder is 128. Positional encoding is used for both encoder and decoder in our system. We use the padding mask of decoder to mask the $<$ pad $>$ in word vectors. Cross entropy function is used as the loss function, which ignores the index of $<$pad$>$.

Our model is trained by Adam optimizer. The initial learning rate is set as 0.0001, and cosine warm-up strategy is adopted to adjust the learning rate. The batch size is set as 128 and the max training epoch is set as 300. Early stopping strategy is used according to the loss value on the validation set. When the loss value does not decent for continuous 20 epochs, the model training process will be stopped. The pre-trained encoder is optimized by Adam, too. The loss function of multi-label classification is BCEloss function, and early stopping strategy with patience as 10.

3.3. Performance Comparison

In our experiments, except from the baseline system of DCASE2021 Task6 [3], a Transformer based system, a variation of our proposed system (i.e., our system without time dimension squeeze in audio embedding and pre-trained encoder, denoted as MLP/w/AE) and another variation (i.e., our system without pre-trained encoder, denoted as MLP/w/Pre) are also provided for performance comparison. Note that, the Transformer based system is built upon the Transformer model [10] without audio embedding process and pre-trained encoder.

The experimental results are given in Table 2. As can be seen from Table 2, the Transformer based system, MLP/w/AE, MLP/w/Pre and our proposed system all outperform the baseline system in terms of all performance metrics. Compared with the Transformer based system, the MLP/w/AE has slightly higher BELU, ROUGE$_L$ and SPICE metric scores and slightly lower CIDEr and SPIDEr scores. Overall, they achieve similar performance. This shows that the self-attention strategy is not necessary for the encoder of an audio captioning system, and the use multilayer perceptrons can achieve comparable or even better performance to the self-attention module. Compared with MLP/w/Pre, though the system with pre-trained encoder has slightly lower BELU, ROUGE$_L$ and METEOR scores, it can achieve higher CIDEr, SPICE and SPIDEr scores, the reason is that the pre-trained encoder can provide more content information of audio signal while reducing the probability of the function words. With audio embedding process and pre-trained encoder, our system can achieve the best performance in terms of CIDEr, SPICE and SPIDEr.

Table 3: Number of parameters of different systems (M: million)

Baseline	Transformer	MLP/w/AE	Proposed system
4.57M	2.52M	3.97M	2.45M

We also give the number of parameters of different systems in Table 3. Note that, the MLP/w/Pre has the same number of parameters, so we do not show its. We can see that our proposed system has the least amount of parameters as compared with other systems. According to Table 2 and Table 3, our system can achieve the best performance with less parameters, which shows the positive influence of the audio embedding process and the effectiveness of MLP-Mixer for ACC task.

4. CONCLUSION

In this technical report, we give our AAC system for Task6 of the DCASE2021 challenge in detail. Our system is a sequence-to-sequence autoencoder model, which includes an encoder based on MLP-Mixer and a self-attention decoder based on Transformer. The experimental results show our system can achieve best performance with the least amount of parameters as compared with other systems. In future work, we will try use other encoders or decoders structures with other pre-trained model for the automated audio captioning task.

5. ACKNOWLEDGEMENT

This research was supported by the Natural Science Foundation of Heilongjiang Province under Grant No. YQ2020F010.

6. REFERENCES

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

[2] S. Ikawa and K. Kashino, "Neural audio captioning based on conditional sequence-to-sequence model," 2019.

[3] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

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

[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[6] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[7] D. Rothman, "What's wrong with CNNs and spectrograms for audio processing?" *Tech. Rep.*, 2018.

[8] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 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.

[9] I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, D. Keysers, J. Uszkoreit, M. Lucic, *et al.*, "Mlp-mixer: An all-mlp architecture for vision," *arXiv preprint arXiv:2105.01601*, 2021.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS*, 2017.

[11] P. S. Ow and T. E. Morton, "Filtered beam search in scheduling," *The International Journal Of Production Research*, vol. 26, no. 1, pp. 35–62, 1988.

[12] Y. Wu, K. Chen, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pre-training for 2020 DCASE audio captioning challenge," DCASE2020 Challenge, Tech. Rep., June 2020.