# AUDIO CAPTIONING USING PRE-TRAINED MODEL AND DATA AUGUMENTATION

Technical Report

*Tianyang Huang[1], Chaofan Pan[1], Wenyao Chen[1], Chenyang Zhu[3], Shengchen Li[2], Xi Shao[1]*

[1] College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China, {1220013304, 1021010410, 1221013730} @njupt.edu.cn
shaoxi@njupt.edu.cn

[2] School of Advanced Technology, Xi'an Jiaotong-liverpool University, Suzhou, China,
Shengchen.Li@xjtlu.edu.cn

[3] School of Artificial Intelligence and Computer Science, Jiangnan University,
Wuxi, China, chenyangzhu2018@163.com

## ABSTRACT

This technical report describes an automatic audio captioning system for task 6, Detection and Classification of Acoustic Scenes and Events (DCASE) 2022 Challenge. Based on an encoder-decoder architecture, the system is composed of convolutional neural network (CNN) encoder and transformer decoder. Instead of using pre-trained models only in audio modal, we try to introduce pretrained models in text modal as well. In addition, we consider using a data argumentation method with and without noise to improve the data quality and thus improve the generalization and robustness of the model. The experimental results show that our system can achieve the SPIDEr of 0.257 (official baseline: 0.233) on the Clotho evaluation set.

***Index Terms—*** Automated audio captioning, sequence-to-sequence model, data augmentation, cross-modal task

## 1. INTRODUCTION

Automatic audio captioning (AAC) is a multi-mode task that translates input audio into corresponding descriptions (i.e., captions) using natural language [1]. This task expects the caption to be as comprehensive as possible. Different from acoustic event detection (AED) and acoustic scene classification (ASC), AAC aims to describe general information, including the recognition of acoustic events, sound scene, foreground recognition, background recognition, and concepts of objects or environments [2]. AAC has positive effects on various applications, such as helping hearing-impaired people understand sounds in surrounding environments, analyzing sound in smart cities for security monitoring and intelligent and content-oriented machine-to-machine interaction [3].

The encoder-decoder architecture with CNN-Transformer was shown to give excellent performance in the DCASE 2021 challenge and thus is chosen as the baseline system in our Work [4]. The architecture in this report consists of an encoder and decoder, which is a sequence-to-sequence model. The encoder is formed by a 10-layer CNN [5], extracting audio features. The decoder is a 4-layer Transformer decoder, receiving the audio features extracted from audio modal by the encoder and the word embeddings extracted from text modal for language generation [6]. Training an end-to-end audio captioning system from scratch becomes more difficult when only a small amount of data is available. The pre-trained CNN layers are adopted from a CNN based neural network for acoustic event tagging, which makes the latent variable resulted more efficient on generating captions. On the basis of using a pre-trained model, we adopt additional data argumentation methods and try different data enhancement methods to carry on the research.

This report describes the methods we submitted to Subtask A, Task 6 of DCASE 2022 challenge. Our systems are based on a sequence-to-sequence framework, formed by a CNN encoder and a Transformer decoder, with additional improvements, including the use of data argumentation methods and different optimizers.

The organization of this report is organized as follows: Section2 describes the structure of our AAC system. Section 3 presents the details of experiments and results. Section 4 concludes our work.

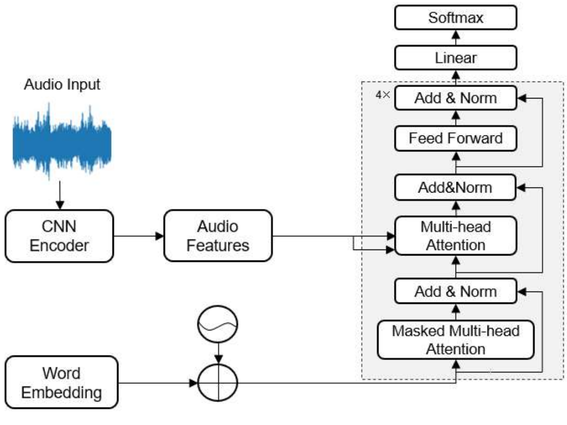

Figure 1: Architecture of the proposed model

Table 2: Performance comparison on Clotho dataset.

| Model | BLUE$_1$ | BLUE$_2$ | BLUE$_3$ | BLUE$_4$ | ROUGE$_L$ | METERO | CIDEr | SPICE | SPIDEr |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.555 | 0.358 | 0.239 | 0.156 | 0.364 | 0.164 | 0.358 | 0.109 | 0.233 |
| PANNs | 0.560 | 0.363 | 0.240 | 0.156 | 0.375 | 0.169 | 0.381 | 0.117 | 0.249 |
| PANNs-WE-Adam | 0.562 | 0.361 | 0.240 | 0.156 | 0.377 | 0.171 | 0.384 | 0.118 | 0.251 |
| PANNs-WE-AdamW | 0.567 | 0.363 | 0.240 | 0.155 | 0.380 | 0.174 | 0.386 | 0.121 | 0.253 |
| PANNs-WE-Mixture-Noise | 0.568 | 0.367 | 0.245 | 0.161 | 0.383 | 0.175 | 0.395 | 0.119 | 0.257 |

## 2. SYSTEM STRUCTURE

In this section, we describe the architecture of our system, including the CNN encoder and transformer decoder.

### 2.1. CNN Encoder

In order to get audio features, we choose CNN as the encoder. Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition (PANNs) has been proved effective in extracting audio features, so we applied the parameters provided by it to initializing the parameters of CNN.

Due to the local feature extraction capability, CNN is applied to extract features of input audios. Besides, as presented in previous work, 10-layer CNN is more effective at preventing overfitting compared to other deeper neural networks. Batch Normalization was used to speed up the training of the model. The number of channels in each convolutional block is 64, 128, 256, 512, respectively.

### 2.2. Transformer Decoder

For the decoder of the model, we select Transformer, which is good at handling various tasks in Natural Language Processing (NLP), to connect the audio features with word embeddings. We only used the decoder part of the Transformer.

A standard Transformer structure consists of an encoder and a decoder. Our input is an audio feature, whose length is far more than a sentence, and the Transformer's encoder is not applicable. Therefore, our model only uses the decoder part of the Transformer to generate annotation statements for audio features. The decoder consists of 4 layers with 4 heads and the dimension of the hidden layer is 128.

In the decoder-only Transformer’s second multi-head attention block, audio features construct a connection with word embeddings. By the way, the feature space of the audio mode and the feature space of the text mode are mapped.

## 3. EXPERIMENTS AND RESULTS

### 3.1. Data Pre-processing

Our system works on the Clotho (v2) dataset from Task6 of DCASE2022. Clotho dataset consists of audio samples of 15 to 30 seconds duration, each audio sample having five captions of 8 to 20 words length. There is a total number of 6,972 audio samples in Clotho, with 34,860 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 44.1kHz. We get 64 log-mel band spectral, using 1024 points Hamming window with 50% overlap. To unify the encoder dimensions, we pad the audio spectral to the max time sequence length with 0. All captions in the dataset are transformed to lower case with punctuation removed. Two special tokens “<sos>” and “<eos>” are used to pad the caption.

### 3.2. Experimental Setup

The whole model is trained with a batch size of 16. Warm-up is used in the first 5 epochs for the learning rate linearly increased from 0 to 0.001. The learning rate is decreased to 1/10 of itself every 5 epochs after the warm-up. To deal with over-fitting problems, dropout with rate of 0.2 is applied in the proposed model. Label smoothing is applied in all the experiments to improve the generalization ability of the model [7]. SpecAugment about data argumentation methods is used in two different making types, “zero-value” and “mixture” [8]. Zero-value, which directly masks consecutive time frames and frequency channels with zeros. Mixture utilizes the time frames and frequency channels of other samples within the mini-batch for masking. These methods can be seen as introducing additional noises generated from the dataset, and guide the networks to be more discriminative for classification. In “mixture”, we additionally add an extra gaussian noise. The Beam Search algorithm is to find the optimal solution in the relatively limited Search space with less cost, and the solution is close to the optimal solution in the whole Search space. It helps model achieve a better output. During the inference stage, a beam search with a beam size of 3 is used to improve the decoding performance.

We used two different optimizers for these four experiments. PANNs, PANNs-WE-Adam and PANNs-WE-Mixture-Noise were trained on the Adam optimizer. The AdamW optimizer was additionally tried in PANNs-WE-AdamW and all other settings are same as in PANNs-WE-Adam.

### 3.3. Performance

Our submission contains the following four models:

- PANNs: In this model, audio modal adopts pre-training model PANNs.
- PANNs-WE-Adam: In this model, text modal adopts pre-training model Word2vec, and all other settings are the same as the first model.
- PANNs-WE-AdamW: In this model, we try a new optimizer AdamW, and all other settings are the same as the second model.
- PANNs-WE-Mixture-Noise: The model adopts the data argumentation, “mixture”, different from “zero-value” adopted in models above. It additionally adds an extra gaussian noise.

The performances of our submitted models are shown in Table 1. As can be seen in Table 1, Our model shows better performance than official baseline, which also adopts audio modal pre-training. Just adding text modal pre-training does not provide a good optimization effect. While in the system with reinforcement learning, add text modal pre-training can significantly improve the performance [4]. It proves that simply adding text modal pre-training does not necessarily result in the same improvement as audio modal pre-training. We find that use AdamW as optimizer can achieve better performance than traditional Adam optimizer. In PANNs-WE-Mixture-Noise, we equip mixture with a gaussian noise and get achieve the SPIDEr of 0.257 (official baseline: 0.233) on the Clotho evaluation set.

## 4. CONCLUSION

This technical report primarily describes our system and the approach to Task 6, Subtask A in 2022. In addition to inheriting the pre-training methods from the previous two years of popular transfer learning, we have additionally improved the data argumentation technique and tried to introduce noise into the data argumentation to improve the generalization ability and robustness of the model. In addition, switching to the AdamW optimizer has improved performance in our experiments. In future work, we will try use other encoders or decoders structures and apply more advanced data argumentation methods into them.

## 5. 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.* IEEE, 2017, pp. 374–378.

[2] 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.

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

[4] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. L. Tang, X. Shao, M. D. Plumbley, and W. Wang, “An encoder-decoder based audio captioning system with transfer and reinforcement learning for DCASE challenge 2021 task 6,” DCASE2021 Challenge, Tech. Rep., July 2021.

[5] 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,” *arXiv preprint-arXiv:1912.10211*, 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones,A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[7] 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.

[8] H. Wang, Y. Zou, and W. Wang, “Specaugment++: A hidden space data augmentation method for acoustic scene classification,” *arXiv preprint arXiv:2103.16858*, 2021.