THE DCASE 2021 CHALLENGE TASK 6 SYSTEM: AUTOMATED AUDIO CAPTIONING WITH WEAKLY SUPERVISED PRE-TRAING AND WORD SELECTION METHODS

Technical Report

Weiqiang Yuan, Qichen Han*, Dong Liu, Xiang Li, Zhen Yang*

NetEase (Hangzhou) Network Co., Ltd., China,
{yuanweiqiang, hanqichen, hzliudong, yangzhen1, hzlixiang}@corp.netease.com

ABSTRACT

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. We use encoder-decoder modeling framework for audio understanding and caption generation. Our solution focuses on solving two problems in automated audio captioning: data insufficiency and word selection indeterminacy. As limited audios with golden captions are available, we collect large-scale weakly labeled dataset from Web with heuristic methods. Then we pre-train the encoder-decoder models with this dataset followed by fine-tuning on Clotho dataset. To solve the word selection indeterminacy problem, we use keywords extracted from captions of similar audios and audio event tags produced by pre-trained models to guide words generation in decoding stage. We tested our submissions using the development-testing dataset. Our best submission achieved 31.8 SPIDEr score where that of the baseline system is 5.4.

Index Terms— Audio captioning, encoder-decoder-modeling, weakly supervised pre-training, audio similarity, audio event tag, audio retrieval

1. INTRODUCTION

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 [1]. The automated audio captioning (AAC) problem is defined as an intermodal translation task of automatically generating a textual description for an input audio signal [2]. This task need information includes identification of sound events, acoustic scenes, spatio-temporal relationships of sources, foreground versus background discrimination, concepts, and physical properties of objects and environment [3]. Our system is a sequence-to-sequence model, which contains an encoder based on convolutional neural network (CNN) and a decoder based on Transformer.

Our submission focuses on two issues: The first problem is data insufficiency. Clotho [3] v2 dataset only has 6,974 audios and 34,870 captions, which is difficult to support the training of complex models. It is well-established that pre-training on large datasets followed by fine-tuning on target datasets boosts performance. We use heuristic method to construct a weak supervised dataset for pretraining, which contains 65667 audio and its caption. In addition, our system uses PANN's [4] architecture as encoder, which is an audio neural networks trained on the large-scale AudioSet dataset.

The second problem is word selection indeterminacy. In AAC task, due to the use of natural language, an audio can be described in many possible ways. However, in training data, there are only five possible forms of an audio. Such indeterminacy leads to too large search space and difficult training. Considering that similar audio should have similar expressions, we try two methods to introduce this information into decoding. The first is to introduce audio event tags information to assist decoding. The second is based on audio retrieval method. We train a model to calculate the similarity between audios, and use these keywords extracted from the captions of similar audios to assist decoding or model fusion.

Experiment results show that the proposed method outperforms the previous baseline model and reached a SPIDEr score of 31.8 on the development-testing split of Clotho.

2. SYSTEM DESCAPTION

2.1. Data pre-processing

All audio samples are down-sampled at 32 kHz. As for acoustic feature, we use two types of audio features, one is logmel-spectrograms calculated from the time-domain input, the other is time-domain waveforms. Because encoder adopts the pre-training model, the configuration of audio feature extraction is consistent with PANN.

We tokenize the captions with a one-hot encoding of the words, and add <UNK> marks to meet the needs of data augmentation. <SOS> and <EOS> are also employed as the start-of-sequence and end-of-sequence tokens, respectively.

* These authors contributed equally to this work.

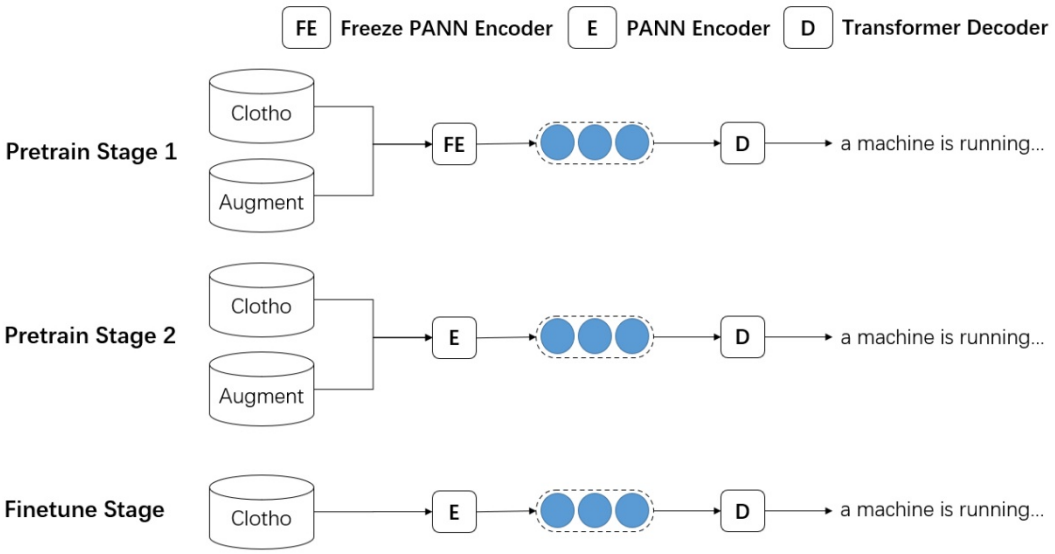

Figure 1: The overview diagram of the proposed model

2.2. Data augmentation

Aiming at the problem of insufficient data, development-validation is added to the development-training set. Noise is added to the audio data from the angle of speed, volume and reverberation, and expand the audio data to 5 times of the original. In addition to Clotho, we also we add the AudioCaps [5] dataset into training set.

To further expand the data, we crawl audios and its description from the Freesound[1], Zapsplat[2], Soundbible[3] and SoundJay[4] website. For audio data, keep the audio longer than five seconds. For the audio longer than 30s, we randomly select a value between 15 and 30 for the duration of the segments and cutting a segment from the rest. We use some heuristic rules to clean and select the descriptions. The mainly rules are as follows:

- Replace non-English words with white space.
- Remove special pattern words such as “record of”, “recorded from”.
- Drop descriptions contain numbers as most of these are equipment types.
- Drop descriptions contain personal pronouns words such as “I, we, my, me”.
- Drop descriptions contain words which do not descript contents of audio but time, places, record methods, equipment, post process and so on.
- Drop descriptions which are shorter than 5 or longer than 20.
- Drop descriptions in which the rate of UNK words according to vocabulary collected from captions in Clotho dataset is bigger than 0.2.

Using the above method, we extracted 65667 training data from four websites. The total number of expanded datasets is 117227 after merging the AudioCaps datasets.

2.3. E2E caption generation

The basic structure of the whole E2E is shown in the figure 1. The whole training process is divided into three stages. In the first stage, the pre-training parameters of encoder are frozen and only the decoder part is trained, In the second stage, the encoder unfreezes and participates in the training, In the third stage, only the Clotho data set is reserved for training.

Encoder In order to fuse the models, three different networks in PANNs are selected for training, namely CNN14, Resnet38 and Wavegram-Logmel CNN[5]。 CNN14 consist of 6 convolutional blocks and each convolutional block consists of 2 convolutional layers with a kernel size of 3×3. Resnet38 consist of 16 basic blocks in the Resnet [6], where each block consists of two convolutional layers with a kernel size of 3×3, and a shortcut connection between input and output. Wavegram-Logmel-CNN use CNN14 as a backbone architecture on the extracted Wavegram and logmel features, where Wavegram are extracted from time-domain waveforms by one-dimensional CNN followed by three convolutional blocks.

Decoder The decoder used in the proposed model is a standard transformer [7], consist of multi-head self-attention on text sequence and multi-head encoder-decoder attention on extracted feature sequence. The decoder uses a 2-layer Transformer with a hidden dimension of 256 and 4 heads. In order to reduce the search space, we also try to introduce tag assisted decoding in some model. During the training process, the decoder is required to predict the tag of audio before generating caption. In the test phase, only the decoding path containing the corresponding tag of test audio is reserved. Tag system is selected from AudioSet Ontology[6] according to Clotho dataset, including 13 categories, named Self-Tag-13. The 13 categories are "Human sounds", "Source-

[1] https://freesound.org
[2] https://www.zapsplat.com/
[3] https://soundbible.com/
[4] https://www.soundjay.com/
[5] https://zenodo.org/record/3987831#.YMhofqgzaUk
[6] http://research.google.com/audioset/ontology/index.html

ambiguous sounds", "Animal", "Sounds of things", "Music", "Natural sounds", "Channel, environment and background", "Vehicle", "Human voice", "Wild animals", "Domestic sounds, home sounds", "Water", "Motor vehicle (road)". Since Self-Tag-13 is derived from AudioSet ontology, we can use the hierarchical relationship of ontology to construct the mapping relationship from PANN Tag to Self-Tag-13, and then transform the prediction results of PANN into tags in Self-Tag-13. We use the prediction of CNN14[1] to get the Self-Tag-13 tag of all training sets and test sets.

Regulations and other detail To improve performance and avoid over-fitting, we also use Label smoothing [8] and SpecAugment [9]. The configuration of SpecAugment is consistent with that of PANN. The learning rate of the three stages of training is 3e-4, 1e-4 and 5e-5. In the inference stage, a beam search with a beam size of 3 is implemented to achieve better decoding performance.

2.4. Similar audio searching

As mentioned above, similar audios have similar expressions. For an audio without captions, we can get relevant keywords from captions of its similar audios. And these keywords can help generate better caption.

Inspired by text similar calculation methods such as ESIM [10], we design a model to calculate the similar between audios. We use CNN14 as audio encoder and get the 2048-dimension feature sequence. Then two audio feature sequences are fed in ESIM network. And cosine similar between audios are calculated at the last layer of ESIM network.

We train this model with triplet dynamic margin loss. For an anchor audio a, it's similar audio p, and unsimilar audio n, $Loss(a,p,n) = \max(0, m(a,p,n) + s(a,p) - s(a,n))$ Where $s(.)$is the similary calculation function defined by the model mentioned above, and $m(.)$ is the margin function. The margins for each pair of (a, p, n) are different. We calculate the margins by the SPIDEr scores between captions of audios. $m(a,p,n) = \max(0.4, SPIDEr(a,p) - SPIDER(a,n))$.

2.5. Ensemble

We use two methods to fuse the models. The first, and the most common, is to decode directly using the average score of different models. Each model in the ensemble outputted log-probabilities $\ln p(w_x|x, w_1, \dots, w_{n-1})$, and we took the average of all log-probabilities in the beam search phase.

The second is scoring the captions generated by different models with captions of similar audios. Then choose the best one as the final caption. Specifically, we get term-frequency weights of each word with 50 captions of 10 most similar audios. Then for each generated caption, we calculate its score by accumulating the term-frequency weight from all of its words.

2.6. Submitted systems

We used two model ensemble strategies to output the final results. The four submitted results use different training dataset or ensemble strategies of following models.

Table1: model configure

Model	Encoder	Use Self-Tag-13
Model1	CNN14	No
Model2	Resnet38	No
Model3	Resnet38	Yes
Model4	Wavegram-Logmel-CNN	No

The details of four submitted systems are followings:

Submission 1 Ensemble of 3 for each Model, 12 models in total. The model is trained by using the development-training and development-validation of Clotho, AudioCaps dataset and the data crawled from Freesound. Then beam ensemble is used to fuse the results of same type models, and finally cap ensemble is used to fuse the results between different type models.

Submission 2 Ensemble of 3 for each Model, 12 models in total. The model is trained by using the development-training and development-validation of Clotho, AudioCaps dataset and the data crawled from four website. Then beam ensemble is used to fuse the all results of 12 models.

Submission 3 Ensemble of 3 for each Model, 12 models in total. Based on the dataset of **Submission 1**, eight hundred development-testing audios are added. Then beam ensemble is used to fuse the results of same type models, and finally cap ensemble is used to fuse the results between different type models. Then beam ensemble is used to fuse the all results of 12 models.

Submission 4 Ensemble of 15 for each Model1 and Model4, 30 models in total. Merge all existing data and divide dataset into 5 parts. The model is trained with 5-fold cross-validation. Then, beam ensemble is used to fuse the all results of 30 models.

Table2: Experimental results on development-testing part of Clotho dataset.

Model	B-1	B2	B3	B4	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr
Model 1	58.3	38.8	26.5	17.8	17.9	38.5	47.3	12.8	30.0
Model 2	59.3	40.0	27.4	18.4	18.3	39.2	48.2	13.3	30.8
Model 3	58.1	38.6	26.1	17.3	17.8	38.4	45.6	13.1	29.4
Model 4	58.5	39.2	26.9	18.2	17.7	38.9	47.4	13.0	30.2
Ensemble	60.3	41.4	28.6	19.5	49.9	18.6	40.0	13.7	31.8

[1] https://github.com/qiuqiangkong/audioset_tagging_cnn

3. EVALUATION ON DEV-TEST DATASET

Table 2 demonstrates the performance of our system on the development-testing split of Clotho dataset [2]. The single model and fusion results of **submission 2** are given in this table, which is the result with our best effective score of SPIDEr in the development-testing part of Clotho dataset.

4. CONCLUSIONS

This technical report described the system participating to the DCASE 2021 Challenge Task 6 [1]. Our submission focused on solving the data insufficiency and word selection indeterminacy problems. We create a new weak supervised dataset for AAC task pre-training, and use PANN pre-training model as encoder to solve the problem of data insufficiency. We solve the problem of word selection indeterminacy by introducing tag information and keywords from audio retrieval in decode stage. The SPIDEr score of our submission on the development-testing dataset is 31.8.

5. REFERENCES

[1] http://dcase.community/workshop2021/.
[2] K. Drossos, S. Adavanne, and T. Virtanen, “Automated Audio Captioning with Recurrent Neural Networks,” in Proc. of IEEE Workshop on Application of Signal Process. to Audio and Acoust. (WASPAA), 2017.
[3] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An Audio Captioning Dataset,” in Proc. of Int’l Conf. on Acoust., Speech, and Signal Process. (ICASSP), 2020.
[4] Kong, Qiuqiang, et al. "Panns: Large-scale pretrained audio neural networks for audio pattern recognition." in IEEE/ACM Transactions on Audio, Speech, and Language Processing 28 (2020): 2880-2894.
[5] C. D. Kim, B. Kim, H. Lee, and G. Kim, “AudioCaps: Generating Captions for Audios in The Wild,” in Proc. of the North American Chapter of the Association for Computational Linguistics: Human Lang. Tech. (NAACL-HLT), 2019.
[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
[7] 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.
[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition, vol. 2016. IEEE, 2016, pp. 2818–2826.
[9] 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,” Proc. Interspeech 2019, pp. 2613–2617, 2019.
[10] Chen, Qian, et al. "Enhanced lstm for natural language inference." arXiv preprint arXiv:1609.06038 (2016).