LABEL-REFINED SEQUENTIAL TRAINING WITH NOISY DATA FOR AUTOMATED AUDIO CAPTIONING

Technical Report

Jaeheon Sim[1⋆], *Eungbeom Kim*[1⋆], *Kyogu Lee*[1,2]

[1] Interdisciplinary Program in Artifical Intelligence, Seoul National University, Seoul, Korea,
[2] Department of Intelligence and Information, AIIS, Seoul National University, Seoul, Korea,
{sjhoney0112, eb.kim, kglee}@snu.ac.kr

ABSTRACT

This technical report describes the submission to the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 Challenge Task 6A: Automated Audio Captioning. We utilize a label-refined sequential training method to leverage the large additional dataset which contains two types of noise including domain shift and label noise. We investigate the usefulness of the additional noisy dataset and observe that the models directly trained on the dataset naively including the additional and target dataset suffer from a poor performance. From this observation, we aim to fully leverage the additional dataset by addressing the two types of noise simultaneously. We sequentially train the model with the prior knowledge about the difference between the target dataset and each of the additional datasets, from the largest to the nearest. We finally train the model on the target dataset, thereby progressively minimizing the domain gap. After this training procedure, we apply a label refinement method which is based on pseudo-labelling from self-training method and repeat the sequential training procedure. The proposed method mitigates the noise in the dataset and achieves the improved performance.

Index Terms— Automated Audio Captioning, Multi-modal Learning, Noisy Labelled Dataset, Semi-supervised Learning

1. INTRODUCTION

In the history of the artificial intelligence, lots of machine learning systems were trained with human-selected hand-crafted features from data that were believed to be able to handle certain tasks. For example, MFCC, one of frequency-based sound features, is used to represent the acoustic feature of human voice in automatic speech recognition tasks and it showed great performances. However, since the improvement of deep learning techniques changed the paradigm from machine learning to deep learning, most of training systems have been developed in an end-to-end fashion, which means a deep learning model learns from raw data and extracts the most helpful features from its own learning mechanism. Thus, the traditional hand-crafted features are not popularly used anymore.

After the transition, many works focus on how to construct a deep learning model that can automatically capture or generate the most helpful learning representation from data; a speech model aims to extract the speech representation, a language model the language representation, and so on. There have been great improvements in such fields by consistently studying how to represent the data of each field in a *uni-modal* fashion. As the deep learning field becomes more mature and more datasets are created, recently, there are attempts to solve *multi-modal* tasks, in which various types of data have to be addressed jointly. On the contrary to the uni-modal learning, where the model discovers the representation that are most related to one type of data, the multi-modal model should consider various types of data and not be biased to certain data types.

Automated audio captioning (AAC) is one of those tasks that have a multi-modality problem. An audio sound has particular features and they can be described with a human language in a textual form. An AAC model has to resolve the relationship between the audio features and its text description. Here we encounter the problem that deep learning models require as much data as possible for performance improvement. Accordingly, a multi-modal model requires multi-modal data and in the case of AAC, the dataset should contain both audio data and its corresponding text data. For this reason, multi-modal tasks like AAC suffer from data shortage.

Thus, in addition to the given dataset, Clotho [1], we collected external audio-text paired datasets: AudioCaps [2],

*Equal contribution. This work was partly supported by Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2022-0-00320, Artificial intelligence research about cross-modal dialogue modeling for one-on-one multi-modal interactions, 90%) and [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University), 10%]

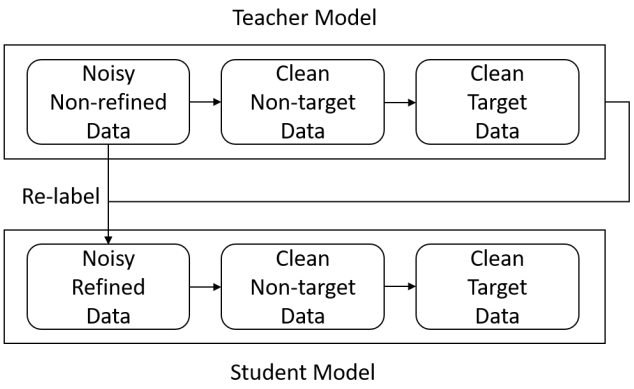

Figure 1: The overall training flow using the sequential training and label refining methods.

WavText5K [3], BBC Sound Effects[1]. The details of each dataset are in 3.2. Clotho and AudioCaps are datasets that were originally created for AAC or audio-text retrieval. Therefore, they have well-preprocessed audio samples and clear text captions (*clean dataset*). However, the other datasets have other purposes for use, so they have different data distribution from Clotho or AudioCaps and include noisy text labels (*noisy dataset*).

To fully exploit the additional data, we adopt a sequential training method, where the datasets are given to the model in pre-defined order. To take one more step to utilize the noisy datasets, we re-label the audio samples by using the pre-trained teacher model and then train a student model with these refined data samples and clean data together. The student model is also trained in the sequential manner. The details of these methods are in 2.2 and 2.3.

2. METHODS

2.1. Automated Audio Captioning

AAC is to translate audio information into a text description. There are two ways to express given audio information in a textual form. 1) *Transcription*: to write down exactly what the sound is. For example, given speech, a model generates words or sentences that are spoken in the speech. 2) *Description*: to write sentences that explains the features of the sound. The main goal of AAC is to generate descriptions from the audio data.

As AAC translates audio sequence into text sequence, most of AAC systems have an encoder-decoder architecture and the system pipeline is as follows:

1. Audio data is fed into the audio encoder.
2. The audio encoder outputs audio feature vectors.
3. The text decoder decodes the audio feature vectors.
4. The text description is generated.

[1]https://sound-effects.bbcrewind.co.uk/

2.2. Sequential Training

Transfer learning is a very well-known learning technique. It leverages a sequential training scheme which pre-train a large-size model on large-scale data to get general representations on certain data modalities and fine-tune the pre-trained model on specific downstream tasks to reduce the domain gap between the large-scale data and the task-specific data [4]. Our proposed method is similar to the transfer learning in that we consider the order of data inputs during training. The difference is that the transfer learning uses the outer datasets for general representation learning, but ours is to tackle the domain difference and noise of the additional datasets(more like parameter initialization rather than knowledge transfer). We propose to train the model in the order of the noisy external dataset (WavText5K, BBC Sound Effects) $\rightarrow$ the clean external dataset (AudioCaps) $\rightarrow$ the target dataset (Clotho). In this way, we can train the model from the largest data(domain shifted) to the nearest data (domain close), thereby naturally and progressively adapt the model to the target dataset.

2.3. Label Refining

The collected external datasets contain noisy text captions, so it is undesirable to use them directly for the training. Thus, we apply a label refinement method which pseudo-labels the noisy data, a method based on self-training. We first train a teacher model with the noisy datasets and the clean datasets using the sequential training method. After the training, the teacher model pseudo-labels the noisy datasets as in the work of [5]. Then, a student model trains with the pseudo-labelled datasets and the clean datasets, also in the sequential manner. The overall architecture can be seen in Figure 1.

3. EXPERIMENT

3.1. Baseline

Audio Encoder The baseline system provided by the challenge adopts PANNs(Pre-trained Audio Neural Networks) [6]. PANNs has multiple CNN layers and is pre-trained on AudioSet [7] dataset to capture the patterns in audio data. Among various verions of PANNs, we use pre-trained CNN14 model that achieves mAP of 0.439 on the AudioSet tagging task like the baseline system.

Text Decoder The text decoder should generate text tokens, so the baseline system adopts a language model called BART [8] as the text decoder. BART is a variation of BERT [9] and is known for being good at text summarization,

Dataset	File Name	Description
WavText5K	censor beep_1337.wav	Have some words you need to censor? Use this censor beeping sound to cover up that fowl language! Requested by Belle Thanks Belle
BBC Sound Effects	nhu05049133.wav	Eastern Meadowlark (Sturnella Magna) - Song medium close-up. Geese and Sandhill Cranes calling in background. NB: Summer. Exact date of recording not known

Table 1: Noisy examples of WavText5K and BBC Sound Effects. The description of the sample from WavText5K seems to explain how to use the sound and that of BBC Sound Effects contains technical information about the sound recorded.

	Train Dataset	B1	B2	B3	B4	M	R	C	S	SD	SD-FL
Baseline	C	**0.583**	**0.385**	**0.259**	**0.168**	**0.177**	**0.385**	0.415	0.122	0.268	0.263
Refined	W, S, A, C	0.562	0.360	0.232	0.159	0.167	0.365	0.395	0.117	0.256	0.253
	W, S, A, C $\rightarrow$ C	0.560	0.361	0.236	0.149	0.174	0.373	0.399	0.121	0.265	0.262
	W, S $\rightarrow$ C	0.560	0.361	0.236	0.149	0.174	0.373	0.399	0.121	0.260	0.257
	W, S $\rightarrow$ A $\rightarrow$ C	0.578	0.381	0.255	0.166	**0.177**	0.379	**0.431**	**0.126**	**0.279**	**0.275**

Table 2: Results for the evaluation split of Clotho. The metric notations are: **B**: BLEU, **M**: METEOR, **R**: ROUGE-L, **C**: CIDEr, **S**: SPICE, **SD**: SPIDEr, **SD-FL**: SPIDEr-FL. The train dataset notations are: **C**: Clotho, **W**: WavText5K, **S**: SoundDescs(BBC Sound Effects), **A**: AudioCaps. The datasets connected by comma(,) are trained simultaneously by random sampling. $\rightarrow$ symbol means the dataset in the latter part is fine-tuned after pre-training. **Refined**: the caption refinement is applied to WavText5K and SoundDescs only; AudioCaps and Clotho are used as the original.

which is similar to AAC in that an AAC model summarizes the contents of audio data. Unlike the audio encoder, however, we did not use any pre-trained weights for the BART model.

All our settings do not deviate from the challenge baseline system because our proposed methods mainly focus on data utilization.

3.2. Dataset

One of key components in improving a deep learning model is to collect as much data as possible. Therefore, we collected open datasets from the web that have audio-text pair data.

- **Clotho** The official dataset provided by the DCASE challenge. It contains about 5K of audio clips and each audio clip has five corresponding captions.
- **AudioCaps** As its name represents, AudioCaps was created for the captioning task. It contains 46K of audio clips and text captions.
- **WavText5K** WavText5K is a web-crawled dataset created for audio-text retrieval system and audio-text multimodal representation learning. It has about 4K of audio clips and text descriptions.
- **BBC Sound Effects(SoundDescs)** A collection of sound effects that were used in radio or TV shows and information about the sound recorded. It has about 33K of audio clips and text descriptions.

Clotho and AudioCaps have well-preprocessed audio samples and clear text captions because they were intentionally created for the AAC task. Also, we can see in detail how the audio samples were collected and the text captions were annotated in their works [1, 2]. On the other hand, even though WavText5K has a purpose of audio-text representation learning for audio-text retrieval, it uses the data just as it was crawled from the web. BBC Sound Effects has both audio and text data but the text description contains technical information about the sound. Thus, these datasets are too noisy to be directly used in AAC. We can see the examples of noisy text descriptions in Table 1.

In addition, many of the additional audio samples are too long compared to Clotho and AudioCaps. Thus, we removed the data with an audio sample longer than 30 seconds and finally acquire approximately 57K of additional data samples except for Clotho.

3.3. Training

As our work mainly focus on leveraging additional data, our training settings are almost same with the provided baseline system[2]. Also, we increase the number of training epochs to deal with the larger dataset size. In our sequential training setting, our model is firstly pre-trained on the noisy external data and then fine-tuned on AudioCaps. Finally, the model is fine-tuned on the target dataset, Clotho. In each training

[2]https://github.com/felixgontier/dcase-2023-baseline/tree/main

phase, we train the model for 100 epochs with batch size of 128 on NVIDIA GeForce RTX 3090 and pick the model of the best performance step for each training phase.

4. RESULTS

The result in Table 2 shows that by simply training the model in a proper order, the AAC performance can be improved without changing model architectures at all. In line 3 in the table, we can observe that naively including all the additional data for training do not draw any improvements from the baseline. Even when the model is further fine-tuned on Clotho (line 3), the scores stay below the baseline performance. In line 5, however, when the model is sequentially trained on the noisy external datasets, on the clean dataset, AudioCaps, and finally on the clean target dataset, Clotho, it shows huge performance improvements on the main evaluation scores (SPIDEr, SPIDEr-FL) with only slight degradation on the other scores (BLEU, METEOR, ROUGE-L). In addition, as we do not apply reinforcement learning, ensembling or any other data augmentation methods, our model has a potential to achieve much higher scores.

An interesting point is the existence of an intermediate dataset to bridge the noisy dataset to the target dataset. Although the official dataset for AAC in the DCASE task 6A is Clotho, AudioCaps is used as well as Clotho in audio-text multi-modal learning tasks. Thus, AudioCaps also has capability to make good multi-modal representation between audio and text. In line 4 of Table 2, the model achieves much lower scores on all the metrics when directly fine-tuning on Clotho. By inserting AudioCaps between the other datasets and Clotho during training, we could get the best scores as in line 5. We submit this model as our final submission.

5. CONCLUSIONS

In this technical report, we describe our submission to DCASE 2023 Challenge Task 6A: Automated Audio Captioning. We apply a label-refined sequential training method to exploit large additional datasets. Our method is very simple and improves the model performance from the baseline performance. Based on our method, we expect to achieve higher scores by applying other widely-used methods.

6. REFERENCES

[1] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," 2019.

[2] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132. [Online]. Available: https://aclanthology.org/N19-1011

[3] S. Deshmukh, B. Elizalde, and H. Wang, "Audio retrieval with wavtext5k and clap training," *arXiv preprint arXiv:2209.14275*, 2022.

[4] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[5] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] 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," 2020.

[7] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[8] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," 2019.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.