HYU SUBMISSION FOR THE DCASE 2023 TASK 6A: AUTOMATED AUDIO CAPTIONING MODEL USING AL-MIXGEN AND SYNONYMS SUBSTITUTION

Technical Report

Jae-Heung Cho[*], *Yoon-Ah Park*[*], *Jaewon Kim*[*], *Joon-Hyuk Chang*

Department of Electronic Engineering, Hanyang University,
Seoul, Republic of Korea

ABSTRACT

This paper presents the automated audio captioning model for participating in the detection and classification of acoustic scenes and events 2023 challenge task 6A. The model consists of two parts: an audio feature extractor and a language model. The audio feature extractor employed in our model is the pre-trained convolutional neural network 14 (CNN14), trained with AudioSet. CNN14 has demonstrated excellent performance in extracting audio features. For the language model, we utilized bidirectional autoregressive transformers model, which has achieved remarkable success in generating the text. We pre-trained the model with WavCaps, AudioCaps and Clotho dataset to manage the limitation of data availability, and then fine-tuned with Clotho dataset. Furthermore, AL-MixGen and synonyms substitution methods were also implemented for data augmentation. To improve the evaluation metric directly, we trained the model with reinforcement learning to optimize the CIDEr score. Finally, we achieved improved performance by adapting an ensemble of higher-performance models, leading to the accomplishment of 0.343 SPIDEr score.

Index Terms— Audio captioning, pre-training, data augmentaion, reinforcement learning

1. INTRODUCTION

Automated audio captioning (AAC) is an audio to text generation task that combines audio processing and natural language processing to describe audio clips using natural language. Unlike audio event detection and audio classification tasks, AAC aims to capture spatio-temporal relationships in audio clips and perform advanced interpretation of audio. The detection and classification of acoustic scenes and events (DCASE) challenge has played a significant role in promoting research on AAC, particularly with the use of audio-caption pair datasets like Clotho [1] and AudioCaps [2].

During the initial development of AAC models, recurrent neural network (RNN)-based approaches [3, 4, 5] were commonly proposed. However, as attention-mechanism language models [6] with superior performance emerged, transformer-based models gained significant popularity. Various transformer-based architectures, including convolution neural network (CNN)-transformer [7, 8], transformer, and CNN-RNN-transformer [9] with encoder-decoder structures, were widely adopted. These models establish a crucial connection between audio and transformer-based language models. CNN-based encoders have particularly demonstrated outstanding performance in audio representation as audio feature extractors. This combination of transformers and CNN-based encoders has significantly advanced the field of AAC. One of the main challenges with the AAC task has been limited quantity of available dataset. Several approaches have been proposed to address this issue, including utilization of the Freesound dataset [10]. However, the generated captions often fell short of accurately describing the audio, thereby impeding the learning process and limiting the performance of AAC models. To overcome this challenge, the WavCaps dataset [11] has been recently released. The dataset provides high-quality captions that were created with the assistance of ChatGPT, leveraging audio-text pairs from previously underutilized datasets like Freesound and BBC sound effects. The WavCaps dataset aims to provide more precise and informative captions, enhancing the training and performance of AAC models.

The first major component of our model is an audio feature extractor, where we initialize the parameters using the pre-trained audio neural networks (PANNs) [12]. PANNs are pre-trained using the AudioSet audio tagging dataset. This audio feature extractor extracts features from the log mel-spectrogram of the audio clip, which serves as input to our model. The next component of our model is a language model, where we utilize bidirectional auto-regressive transformer (BART) [13]. Transformer-based language models, such as BART, have outperformed CNN and RNN-based models in various tasks. Therefore, we leverage the capabilities of BART to generate text captions based on the extracted audio features. We also incorporated three data augmentation methods: SpecAugment [14], audio language mix generation (AL-MixGen) [15], and synonyms substitution. SpecAugment is a widely used technique that applies random transformations to the log mel-spectrogram of the audio input, aiding in robustness and generalization. MixGen, originally used in image captioning [16], was first implemented in AAC by E. Kim *et al*. Inspired by their work, we observed significant improvements in our model by employing AL-MixGen during the pre-training phase. Additionally, to enhance the model's universality and prevent overfitting during fine-tuning, we conducted synonyms substitution. This involved substituting a random word with synonyms within the caption.

2. SYSTEM DESCRIPTION

2.1. Audio feature extractor

We employed a 14-layer CNN obtained from PANNs to extract audio features in our model. This particular CNN architecture is renowned for its capability to effectively capture audio representations. CNN 14 consists of six convolutional blocks, where each block incorporates two convolutional layers utilizing a 3 × 3 ker-

[*]: Equal contributions.

nel size. After each convolutional layer, batch normalization [17] is applied to standardize the input, and a rectified linear unit (ReLU) activation function [18] is used to enhance the performance. The input to this feature extractor $\mathbf{A} \in \mathbb{R}^{F \times T}$ is the log mel-spectrogram of the audio clip, which is represented by T frames and F filters.

$$\tilde{\mathbf{A}} = Enc(\mathbf{A}). \tag{1}$$

2.2. Language model

We employed the BART as our language model due to its impressive performance on text generation tasks. The BART model architecture consists of an encoder and a decoder, each consisting of 12 transformer layers. The BART encoder takes the audio features generated by the audio feature extractor $\tilde{\mathbf{A}}$ as the input, while the BART decoder takes the output of the BART encoder and the reference caption as the input. The attention mechanism is applied to establish a connection between the BART encoder and the BART decoder, enabling the model to capture the semantic and contextual information of the input sentence. Within each transformer block of the decoder, self-attention is utilized to model the interactions among all the words in the input sentence. This allows the model to generate accurate predictions for the next word, resulting in high-quality text generation. The use of self-attention helps the model capture long-range dependencies and contextual relationships between words.

2.3. Data augmentation

In order to improve the model's generalization properties and attain model consistency between the WavCaps and Clotho datasets, we adapted three different data augmentation techniques: SpecAugment, AL-MixGen, and synonyms substitution. In this section we introduce AL-Mixgen and synonyms substitution among the data augmentation methods.

2.3.1. AL-MixGen

AL-MixGen is a simple but effective multimodal data augmentation technique, which mixups the two audio clips and concatenate the captions.

$$\hat{\mathbf{a}} = \sum_{i=0}^{N} \lambda_i \mathbf{a}_i, \tag{2}$$

$$\hat{\mathbf{t}} = Concat(\mathbf{t}_{i=0}^{N}), \tag{3}$$

where the $\mathbf{a}$, $\mathbf{t}$, $\hat{\mathbf{a}}$, and $\hat{\mathbf{t}}$ denote a waveform of audio, caption, augmented audio, and augmented caption, respectively. $\lambda_i \in [0, 1]$ for i = 0, 1, ..., N is a hyperparameter.

Data augmentation techniques in the multimodal field can present challenges, but AL-MixGen offers a straightforward and effective solution for increasing the amount of data. We employed AL-MixGen to enhance the characteristics of the WavCaps dataset. WavCaps significantly differs from Clotho in terms of audio clip attributes, as it mainly comprises short audio clips with a single event. This discrepancy is a key factor contributing to the substantial performance improvement achieved through AL-MixGen. By introducing overlapping sound events from various sources, AL-MixGen bridges the gap between the limited diversity of each single audio clip in WavCaps dataset and the broader range of audio scenarios present in real-world situations. This augmentation technique plays a crucial role in enhancing the model's performance by capturing and generating captions for a wider array of sound events.

2.3.2. Synonyms substitution

Synonyms substitution is an easy and simple data augmentation technique [19] based on wordnet-based synonym substitution. It involves replacing a specific word in a sentence with another word that has a similar meaning. This technique is employed to enhance the model's ability to generate captions with improved universality. During the refinement learning process, we randomly selected single words from the target captions and replaced them with their synonyms. This approach proved to be beneficial during the fine-tuning phase, contributing to enhanced model performance and reducing overfitting to specific training sets within Clotho.

3. EXPERIMENTS

3.1. Training

Our learning process consists of three steps: pre-training, fine-tuning, and reinforcement learning. In pre-training process, we used WavCaps, AudioCaps, and Clotho. Afterward, we froze the encoder and fine-tuned with the Clotho dataset. In both processes, the cross entropy was used as the loss function.

$$\mathcal{L}_{CE} = -\frac{1}{T}\sum_{t=1}^{T} logp(y_t | y_{1:t-1}, \mathbf{x}), \tag{4}$$

where $\mathbf{x}$, y_t, and T are input audio clip, t-th ground truth token in a sentence, and the length of sentence, respectively. After fine tuning process, the model was fine-tuned by self-critical sequence training (SCST) [20].

3.2. Dataset

3.2.1. WavCaps

The WavCaps dataset[1] is introduced as the large-scale weakly-labelled audio captioning dataset, consisting of approximately 400 k audio clips with paired captions. The WavCaps dataset is obtained from various sources including BBC sound effects, FreeSound, SoundBible and AudioSet. To overcome the issue of noisy and unsuitable raw descriptions, a three-stage processing pipeline using ChatGPT is proposed. The average audio length is 67.59 seconds, while the average text length is 7.8 tokens. Due to the unavailability of certain data from FreeSound, we solely utilized the publicly available data for our study.

3.2.2. AudioCaps

AudioCaps consists of 46 k audio clips paired with text descriptions and the duration of the audio clip is 10 seconds. It is divided into three sets, development-training, development-validation, and development-testing, each of which contains 38,118, 500, and 979 audio clips, respectively. The captions in the train set are single captions per audio clip, while the validation and test sets have five captions per audio clip.

3.2.3. Clotho

The Clotho v2.1 consists of three subsets in the published development sets, development-training, development-validation,

[1] `https://github.com/XinhaoMei/WavCaps`

Table 1: Performances of proposed methods and baseline on Clotho evaluation split. For all metrics, higher values indicate better performance.

Model	BLUE-1	BLUE-4	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr	SPIDEr-FL
Baseline	0.585	0.166	0.177	0.387	0.420	0.119	0.270	0.261
Submission 1	0.600	0.173	0.188	0.394	0.483	0.137	0.310	0.307
Submission 2	**0.678**	0.188	0.195	0.419	0.526	0.143	0.335	0.225
Submission 3	0.676	0.194	0.195	**0.424**	0.539	0.143	0.341	0.233
Submission 4	0.656	**0.202**	**0.197**	0.422	**0.541**	**0.146**	**0.343**	**0.313**

and development-testing. The development-training subset contains 3,839 audio clips, while the development-validation and development-testing subsets consist of 1,045 audio clips each. Each audio file in the dataset has a duration of 15 to 30 seconds. For each audio clip, there are five captions provided, ranging from 8 to 20 words in length.

3.3. Experiment setup

We used a 64 mel-bands, sampling rate of 32 kHz, window length of 1024, and a hop size of 320. We conducted pre-training with a batch size of 32, 20 epochs, and a learning rate of 5×10^{-5} using AL-MixGen. For fine-tuning process, we employed synonyms substitution with a batch size of 32, 20 epochs, a learning rate of 5×10^{-6}, and decreased by a factor of 10 every 10 epochs after the warm-up. Subsequently, we applied the SCST method with 80 epochs and a learning rate of 1×10^{-5} for adjusting the CIDEr score. During inference, we utilized beam search with a beam size of 3 to enhance the decoding performance.

4. RESULTS

The experimental results of the submission is shown in Table 1. The details of the submission methods are following:

- **Submission 1**: CNN-BART model with AL-MixGen and synonyms substitution.
- **Submission 2**: CNN-BART model with AL-MixGen, synonyms substitution, and fine-tuned with reinforcement learning.
- **Submission 3**: An ensemble of the 3 best CNN-BART models.
- **Submission 4**: An ensemble of the 6 best CNN-BART models.

5. CONCLUSION

This paper introduced various methods for submitting to the DCASE Challenge Task 6A. We discussed effective approaches for optimizing the AAC model based on WavCaps dataset. Additionally, we demonstrated the effectiveness of our method by achieving 0.343 SPIDEr score through reinforcement learning and ensemble.

6. REFERENCES

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

[2] C. D. Kim, B. C. Kim, H. M. Lee, and G. H. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 119–132.

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

[4] X. Xu, H. Dinkel, M. Wu, and K. Yu, "Audio caption in a car setting with a sentence-level loss," in *Proc. International Symposium on Chinese Spoken Language Processing* (*ISCSLP*), 2021, pp. 1–5.

[5] M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2019, pp. 830–834.

[6] A. Vaswani *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] K. Chen *et al.*, "Audio captioning based on transformer and pre-trained cnn." in *Proc. Detection and Classification of Acoustic Scenes and Events* (*DCASE*), 2020, pp. 21–25.

[8] X. Mei *et al.*, "Diverse audio captioning via adversarial training," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2022, pp. 8882–8886.

[9] A. Ö. Eren and S. Sert, "Audio captioning based on combined audio and semantic embeddings," in *IEEE International Symposium on Multimedia* (*ISM*), 2020, pp. 41–48.

[10] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. 21st ACM international conference on Multimedia*, 2013, pp. 411–412.

[11] X. Mei *et al.*, "Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[12] Q. Kong *et al.*, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition." *IEEE Trans. Audio, Speech, and Language Processing.*, vol. 28, pp. 2880–2894, 2020.

[13] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[14] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[15] E. Kim *et al.*, "Improving audio-language learning with mixgen and multi-level test-time augmentation," *arXiv preprint arXiv:2210.17143*, 2022.

[16] X. Hao *et al.*, "Mixgen: A new multi-modal data augmentation," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 379–389.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International conference on machine learning*, 2015, pp. 448–456.

[18] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[19] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[20] S. J. Rennie *et al.*, "Self-critical sequence training for image captioning," in *Proc. IEEE conference on computer vision and pattern recognition*, 2017, pp. 7008–7024.