PEACS: PREFIX ENCODING FOR AUDITORY CAPTION SYNTHESIS

Technical Report

Timothy Schaumlöffel[1], *Martina G. Vilas*[1,2], *Gemma Roig*[1,3]

[1]Goethe University Frankfurt, Department of Computer Science,
Robert-Mayer-Str. 11-15, 60323 Frankfurt, Germany
schaumloeffel@em.uni-frankfurt.de, roig@cs.uni-frankfurt.de
[2]Ernst Strüngmann Institute for Neuroscience, Deutschordenstraße 46, 60528 Frankfurt, Germany
martina.vilas@esi-frankfurt.de
[3]The Hessian Center for Artificial Intelligence (hessian.AI), Darmstadt, Germany

ABSTRACT

This technical report describes an Automated Audio Captioning system for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2023 Challenge, Task 6a (automated audio captioning). Our approach employs an encoder-decoder architecture, with the encoder utilizing a large contrastive pre-trained HTS-AT capable of handling variable-length audio segments. The decoder is based on the GPT2 model. To incorporate audio into the decoding process, we employ a light mapping network that translates audio representations into a prefix, effectively guiding the decoder's generation process. Given the limited data availability, we pre-train our model on various audio captioning datasets and fine-tune it on Clotho. We reach a SPIDERr-FL score of 29.3 on the evaluation split of the Clotho-v2 dataset.

Index Terms— audio captioning, transformer encoder-decoder, GPT2, pre-training

1. INTRODUCTION

Audio captioning is the intermodal translation task of describing human-perceived audio information using free text. It enables the expression of arbitrary information beyond fixed labels, such as complex scenes of objects or events and their relationship over time. Collecting this data requires expensive and time-consuming human annotation or high-quality web sources. Both factors limit the amount of existing data and motivate an approach for automatic auditory caption generation.

We adopt a sequence-to-sequence framework to address this challenge, employing an encoder-decoder architecture. Due to the limited amount of data and the complexity of the captioning task, we experiment with using pre-trained models. An encoder creates rich audio embeddings, while a decoder generates sequences based on the encoding. We follow the CLIPCap architecture [1], which trains a mapping network to translate the audio representations into a prefix. This prefix serves as a guide for the decoder during the caption generation process. We call our approach PEACS for **P**refix **E**ncoding for **A**uditory **C**aption **S**ynthesis.

2. SYSTEM DESCRIPTION

The method expects a dataset of size N where each data point consists of an audio signal and a textual description $\{a^i, c^i\}_{i=1}^{N}$. The audio signal is embedded using a strong modality-specific encoder to get semantically rich representations. The goal is to train a model with parameters θ that learns the generation of meaningful captions for an audio file. The target caption is transformed into a fixed length sequence of tokens $c^i = c_1^i, \ldots, c_\ell^i$ with length $\ell = 80$. Longer captions are truncated and smaller are padded by using an attention mask to indicate the model which tokens should be attend to. The training objective can be formalized as following:

$$\max_{\theta} \sum_{i=1}^{N} \log p_\theta(c_1^i, \ldots, c_\ell^i \mid a^i) \tag{1}$$

An autoregressive language model is used to generate the captions. It outputs the probability distribution of the next token based on the previous tokens. For this reason, a prefix created from the encoder's embeddings can be integrated to guide the generation process. The objective reformulates to:

$$\max_{\theta} \sum_{i=1}^{N} \sum_{j=1}^{\ell} \log\, p_\theta(c_j^i \mid a^i, c_1^i, \cdots, c_{j-1}^i) \tag{2}$$

We utilize a $\text{GPT2}_{\text{small}}$ [2] pre-trained on text generation as decoder, since it offers strong performance and is publicly available. For the encoder component, we utilize CLAP (Contrastive Language-Audio Pre-Training) [3] with an HTS-AT [4] backbone. CLAP is designed to learn a shared representation between audio signals and auditory captions. It is trained on a large volume of audio-caption pairs using a contrastive loss function. The CLAP model has demonstrated impressive performance across various audio tasks, which we believe can greatly benefit the captioning task at hand.

A mapping network M creates a prefix p of size k from the representations of the encoder. The prefix has the same dimensions as a word embedding of GPT2:

$$p_1^i, \ldots, p_k^i = M\Big(\text{CLAP}(a^i)\Big) \tag{3}$$

We experiment with different prefix sizes and mapping network but found a simple linear mapping and a prefix length of $k = 20$ works best.

Table 1: Scores of the PEACS model on Clotho evaluation split. Pre-training datasets are AudioCaps (AC), MACS (MC), SoundDescs (SD), WavText5k (WT).

ID	Pre-train datasets	Train GPT	METEOR	ROUGE$_L$	FENSE	CIDE$_r$	SPICE	SPIDE$_r$	SPIDE$_r$-FL
I	-		16.0	37.0	45.9	35.1	11.6	23.4	22.9
II	AC, MC, SD, WT		16.5	38.0	46.9	36.5	11.8	24.1	23.9
III	-	✓	17.3	38.3	47.0	39.5	11.8	25.6	25.5
IV	AC, MC	✓	17.9	38.9	49.3	44.3	12.6	28.5	28.4
V	AC, MC, SD, WT	✓	18.3	39.3	49.8	45.4	13.2	29.3	29.2
DCASE Baseline			17.7	-	-	42.0	11.9	27.0	26.1

The models is trained using the cross-entropy loss between the predicted and the true captions:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{j=1}^{\ell} \log p_\theta(c_j^i | \, p_1^i, \ldots, p_k^i, c_1^i, \ldots, c_{j-1}^i). \quad (4)$$

3. EXPERIMENTS AND RESULTS

3.1. Dataset

We collect multiple audio captioning datasets to enrich the training of our method. Among these datasets, Clotho [5], AudioCaps [6], and MACS [7] stand out as high-quality resources, featuring human-annotated data. In contrast, SoundDescs [8] and WavText5k [9] consist of sound effects collected from various web sources, accompanied by captions constructed using provided metadata.

Table 2: Overview of datasets used for training.

Dataset	#Clips	#Captions	Avg. duration (s)
Clotho [5]	5,929	29,645	22.44
AudioCaps [6]	47,151	52,575	9.84
MACS [7]	3,930	17,275	10.88
SoundDescs [8]	33,020	33,020	115.75
WavText5k [9]	4,517	4,517	20.27

3.2. Data Pre-processing

Our system processes audio samples of arbitrary length using the feature fusion mechanism introduced by [3]. The mechanism combines coarse global information with random local information. The full audio clip is downsampled to 10 seconds, serving as global information. Additionally, three random 10-second clips extracted from the beginning, middle, and end of the original clip are utilized as local features.

The audio clips are converted to a mono channel with a sampling rate of 48 kHz. Log mel-spectrograms using a Hanning window of 1024, a hop size 480, and 64-mel bins are extracted for all 10s segments. We apply SpecAugment [10] to the fused spectrogram with two masks per axis during training. We mask 128 frames on the time axis and 16 on the frequency axis.

The captions are transformed to lowercase and punctuations are removed. The text is tokenized using a pre-trained Byte-Pair-Encoding (BPE) with a vocabulary size of 50k. An audio clip with more than one caption is duplicated in one epoch to get all possible information, as the amount of data is limited.

3.3. Training

We divide the training into two parts. First we pre-train our method on an enlarged dataset, and then fine-tune the model on the development splits of Clotho. The audio encoder is frozen, while we experiment with training the decoder.

The training of the model is conducted on a single GPU with mixed precision and a batch size of 64. For optimization, we employ the AdamW optimizer with a weight decay of 0.01. Through small preliminary experiments, we have determined that a lower learning rate yields better results, considering that only the mapping needs to be learned. The learning rate is thus set to 1×10^{-5} and linearly decreases after a warm-up. All models are trained with a maximum of 30 epochs. Due to rapid over-fitting, the early stopping mechanism is introduced, which stops the training process if the validation loss does not decrease within two epochs.

We fine-tune the model for 10 epochs with a learning rate of 1×10^{-6}.

3.4. Decoding

We utilize beam search with a beam size of 8 to generate captions during inference. Upon qualitative analysis, we noted a tendency of the model to generate repetitive phrases. To address this issue, we conducted experiments employing different decoding strategies. Specifically, we explored the following approaches:

- Repetition Penalty (RP): We introduce an exponential penalty on repeated sequences.
- N-gram Prevention: We prevent the decoding of n-grams that have already appeared in the generated output.

We found that combining a repetition penalty of 1.2 and bigram prevention yield the best performance. A comparison of these methods for PEACS trained on setup V can be found in Table 3.

3.5. Results

Table 1 shows the results of our proposed model on the development-evaluation set of Clotho. We investigate different setups varying in the pre-training datasets. Setup I and III are skipping the pre-training step and are directly trained on the Clotho development set. Setup IV is pretrained on *AudioCaps (AC)* and *MACS (MC)*, while setup V also adds *SoundDescs (SD)* and *WavText5k (WT)*. We submit the results of IV and V.

Freezing the decoder results in a lightweight architecture since only the mapping network needs to be optimized. However, this approach leads to decreased performance, highlighting the importance of training the decoder as well. We observe that pre-training

on other datasets is beneficial, even if the captions are noisy. Furthermore, we observe few fluency errors, as the performance of the SPIDER metric is only slightly affected by the fluency error detection. This indicates the advantage of using a large pre-trained language model.

Table 3: Comparison of using a repetition penalty (RP) and bi-gram prevention for different beam sizes. The performance is measured with $SPIDE_r$-FL of PEACS (V) on the Clotho evaluation split.

Beam size	w/o penalty	RP	RP + Bi-gram
1	25.3	25.8	25.2
3	28.0	28.2	28.8
5	28.5	28.5	29.0
8	28.7	29.0	29.2

4. CONCLUSION

This technical report outlines our submission to Task 6a of the DCASE 2023 challenge. We demonstrate the effectiveness of leveraging pre-trained encoder and decoder architectures, as well as employing the prefixing technique to establish its connections. Our method offers the advantages of being lightweight and fast to train. Furthermore, through improvements in the decoding strategy, we surpass the baseline performance in all evaluation metrics.

5. ACKNOWLEDGMENT

The project was partly funded by the German Research Council (DFG) (FOR 5368 ARENA, RO 6458/2-1, RO 6458/2-1).

6. REFERENCES

[1] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," 2021.

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[3] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," 2023.

[4] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," 2022.

[5] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: an audio captioning dataset," in *Proc. IEEE Int. Conf. Acoustic., Speech and Signal Process. (ICASSP)*, 2020, pp. 736–740.

[6] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 119–132.

[7] I. M. Morato and A. Mesaros, "Macs - multi-annotator captioned soundscapes," July 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5114771

[8] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, pp. 1–1, 2022. [Online]. Available: https://doi.org/10.1109%2Ftmm.2022.3149712

[9] S. Deshmukh, B. Elizalde, and H. Wang, "Audio retrieval with wavtext5k and clap training," 2022.

[10] 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," in *Interspeech 2019*. ISCA, sep 2019.