EXPLORING AUDIO CAPTIONING WITH KEYWORD-GUIDED TEXT GENERATION

Technical Report

Dawid Kiciński[*]*, Teodor Lamort de Gail, Paweł Bujnowski,*

Samsung R&D Institute Poland
Artificial Intelligence
Warsaw, Poland
d.kicinski@partner.samsung.com, t.lamort@samsung.com, p.bujnowski@samsung.com

ABSTRACT

This technical report describes our submission to the DCASE 2022 challenge, Task 6 A: automated audio captioning. In our system, we explore the use of pre-trained language models for the audio captioning task. The proposed system is an encoder-decoder architecture consisting of a pre-trained PANN encoder and a GPT2 decoder. Audio embeddings are encoded to language model prompts using a simple mapping network. We further develop our system by employing strategies of guiding the decoder with textual information. We prompt the decoder with keywords extracted from semantically similar audios and use them to choose the best matching caption by their occurrence.

Index Terms— audio captioning, encoder-decoder, audio similarity, keyword extraction

1. INTRODUCTION

In the task of audio captioning, it is crucial to provide a descriptive summary of sound events in the presence of acoustic background. This requires an understanding of the semantics of sound and the ability to comprehend and produce intelligible text.

Solving such a task requires a high degree of expressive power; therefore, we address this challenge by using pre-trained networks for sound comprehension and, following [1], we experiment with the use of pre-trained language models.

The difficulty of this task is also due to the specificity and variability in Clotho's descriptions, i.e., for a single sound example, the captions may describe different scenes that evoke a given sound. To facilitate the model generation process, we experiment with prompting the decoder network with additional keywords.

2. DATA

The training is performed on the provided Clotho dataset [2], as well as AudioCaps[3]. We experimented with the Hospital & Car dataset, where we translated the captions to English with DeepL, but we found that it does not improve results on the Clotho evaluation split. For AudioCaps, we use the standard train-validation split. For Clotho, we train on the development and validation subsets, and validate on the evaluation subset[1].

In an attempt to gather more data, we downloaded around $500\,000$ descriptions from Freesound, and trained a binary classifier to tell them apart from Clotho captions. The classifier was tasked to output 0 for Freesound and 1 for Clotho. The idea was that Freesound labels with outputs close to 1, i.e., the failure points of the classifier, would be useful audio captions similar to those in Clotho. We then downloaded all the samples corresponding to captions with an output score above 0.05, after filtering out those shorter than 3 seconds and longer than 2 minutes. This resulted in around $19\,000$ additional training examples.

The classifier was built by adding three fully connected layers of size 256 on top of the `all-mpnet-base-v2` model from SentenceTransformers[4], with ReLU activation, and a single sigmoid-activated output neuron.

Table 1: Datasets that are used for two-stage training. For the first training stage, the model is trained on concatenated AudioCaps and collected by us samples from Freesound. In the next stage, the model is fine-tuned only on the Clotho development split. In total, $96\,672$ samples were used for developing our system.

	# training samples	# validation samples
Pre-training datasets		
AudioCaps	$45\,797$	2275
Freesound	$16\,968$	1887
Fine-tuning datasets		
Clotho	$24\,420$	5225

3. METHOD

3.1. Architecture

The basis of our system is a standard encoder-decoder architecture. The key idea is to encode audio in a rich embedding space and then have the decoder generate a sequence of words based on this encoding.

Encoder The encoder is the PANNs[5] CNN14 model. Audio embedding is obtained by taking the output from the penultimate layer. This results in a reduction of an entire audio track into a single embedding with a dimension of 2048. The encoder was pre-trained on AudioSet[6] on the multi-label classification task. We believe that such encoding will contain sufficient information about the acoustic scene and audio events occurring in the sound. Since

[*]Dawid Kiciński is the corresponding author.

[1]According to Clotho naming, as opposed to DCASE Challenge naming.

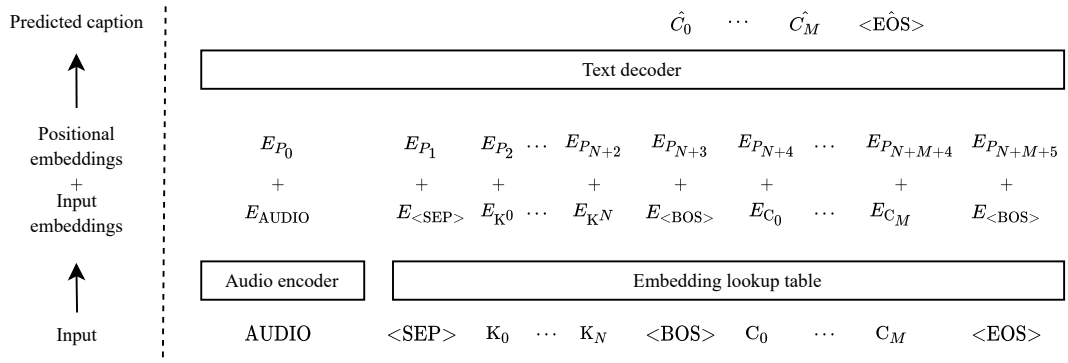

Figure 1: A representation of the multi-modal input to our system. Here, $K_{0...N}$ denotes the indexes of tokenized keywords, $C_{0...M}$ indexes of tokenized captions, and E denotes modality embedding. Audio and text are separately encoded using their respective methods. The positional embeddings are then added to the encoding concatenation. Finally, such a representation is fed into the decoder.

the use of PANNs models has become common in audio pattern recognition and audio captioning, we will omit a background description of the model architecture.

Empirically we found out that it worked best if encoder parameters remained frozen.

Mapping network Models for text generation are traditionally trained using only text modality. This raises a significant challenge with using pre-trained language models - how to translate between different independent latent spaces of models that have not been trained jointly. Following the idea presented in [1], we use a linear mapping layer that maps audio embedding into a single prefix vector of the size of decoder input embedding size. We experimented with different prefix lengths and mapping networks, but this approach proved to be the best.

Decoder In our system we experiment with the classic transformer[7] architectures. We investigated two settings: 1) a GPT2[8] model pre-trained on causal text generation task, and 2) a simple 2-layer transformer with the embedding size of 256 and 8 attention heads. Table 2 compares the number of parameters for the two proposed settings.

Table 2: Number of parameters for two settings that we use during our experiments.

setting	# parameters
w/ 2-layer transformer	104.7 M
w/ GPT2	206.6 M

3.2. Keyword prompting

To reduce the search space for caption decoding, we explore the idea of prompting the decoder network with additional textual information in a form of keywords. We create a database of audio-caption pairs using Clotho development split. For each pair we extract and store a set of a single-gram keywords with KeyBert[9] as well as the pre-computed PANNs embedding. We compare two audios by the cosine distance between two normalized embedding representation.

When a new recording comes in, we search for the best matching audio in our database and use a stored there set of keywords.

3.3. Caption generation

Two text generation methods are used in our system: 1) Beam search with beam size of three 2) Combination of Top-k and Nucleus sampling[10] with keyword guided sample selection. For the list of sampled captions, we select the one that contains the largest number of keywords of the best matching other audio from the training set.

Sampling parameters were selected empirically, k was set to 10, p to 0.70 and number of drawn samples to 20.

3.4. Data preprocessing

To take full advantage of the capabilities of the pre-trained encoder, we keep our audio preprocessing configuration consistent with the PANNs CNN14 model. Audio features are 64-dimensional log-mel spectrograms extracted from resampled to 16 kHz audio.

For captions to match the text preprocessing in evaluation protocol, we remove all punctuation marks, lowercase all letters, and then, depending on the decoder used, we apply a different tokenization scheme. In the case of GPT2, pre-trained BPE tokenization was used, and to meet the expectations of the model expected input, `<endoftext>` tokens were added at the beginning and end of the captions. For transformer trained from scratch, we apply word-level tokenization with a 10 000 token vocabulary constructed from the training data. `<BOS>` and `<EOS>` tokens are added to the caption to indicate the caption's start and end.

In a setting where we additionally prompt the decoder with keywords, we use the `<SEP>` token to separate them from the caption. The model input is constructed by concatenating keywords, audio embedding, and caption. A visualization of this construction can be seen in Figure 1

3.5. Augmentations

The complexity of the task and the scarcity of data make overfitting an issue. Therefore, several data augmentation techniques were applied. Following [11], we apply `Random Crop` and `Random Padding` to keep the audio length under 30 seconds. Then, the white Gaussian noise, whose SNR varies randomly form 10 dB to 120 dB is added to audio with a probability of 0.5.

Finally, the SpecAugment is applied with two masks per axis. We use masks with a maximum width of 64 for the time axis and a width of 8 for the frequency axis.

3.6. Experiments details

In order to take advantage of all the collected data and make our system generate captions that better match those found in Clotho, we used a two-stage training in which we first pre-trained the model on data from AudioCaps and Freesound and then tuned it on Clotho. Table 1 shows the quantification of what data and how much was used in each stage.

In the first phase of training, the AdamW optimizer is used with the default set of parameters except for the initial learning rate, which was set to 1×10^{-4}. After 500 warmup steps learning rate was linearly decreased over 50 epochs Then in the second phase, we fine-tune the decoder in the same training set up with a learning rate set to 5×10^{-5} for five more epochs.

In all our experiments, we use a batch size of 128. To further mitigate overfitting, dropout is set in the decoder to 0.4.

4. EVALUATION

The evaluation results on Clotho development-test dataset for our sumbited systems are given in Table 3. Comparing to DCASE2022 Task 6a baseline system, our solutions score higher on the evaluation metrics.

We observe that 1) using pre-trained language models provides no advantage over training such a model from scratch; 2) our attempts at directing the decoder by giving it a prior in the form of keywords did not work as we expected. We suspect that there was a too weak correlation between keywords and target caption.

Table 3: Scores for evaluation metrics for Clotho evaluation split.

Method	CIDEr	SPICE	SPIDEr
Baseline	0.358	0.109	0.233
GPT2	0.393	0.117	0.255
+ keywords prefix	0.378	0.119	0.249
Transformer	0.433	0.125	0.279
+ guided generation	0.400	0.121	0.260

5. CONCLUSION

This technical report describes our submission for DCASE2022 Task6a challennge in which we investigated the use of pre-trained language-models in the context of audio-captionning as well as utilizing multi-modal prefixes in open-ended text generating. Although our proposed methods did not perform very well, we consider them an exciting starting point that we may develop in the future.

6. REFERENCES

[1] R. Mokady, A. Hertz, and A. H. Bermano, "Clipcap: Clip prefix for image captioning," 2021. [Online]. Available: https://arxiv.org/abs/2111.09734

[2] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," 2019. [Online]. Available: https://arxiv.org/abs/1910.09387

[3] 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

[4] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," 2019. [Online]. Available: https://arxiv.org/abs/1908.10084

[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," 2019. [Online]. Available: https://arxiv.org/abs/1912.10211

[6] 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.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

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

[9] M. Grootendorst, "Keybert: Minimal keyword extraction with bert." 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4461265

[10] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," 2019. [Online]. Available: https://arxiv.org/abs/1904.09751

[11] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending clip to image, text and audio," 2021. [Online]. Available: https://arxiv.org/abs/2106.13043