EFFICIENT AUDIO CAPTIONING TRANSFORMER WITH PATCHOUT AND TEXT GUIDANCE

Technical Report

Thodoris Kouzelis[1,2]*, Grigoris Bastas* [1,2]*, Athanasios Katsamanis*[2]*, Alexandros Potamianos*[1]*,*

[1] Institute for Language and Speech Processing, Athena Research Center, Athens, Greece,
{theodoros.kouzelis, g.bastas, nkatsam}@athenarc.gr
[2] School of ECE, National Technical University of Athens, Athens, Greece,
{potam}@central.ntua.gr

ABSTRACT

This technical report describes an Automated Audio Captioning model for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2022 Challenge, Task 6. We propose a full Transformer architecture that utilizes *Patchout* as proposed in [1], significantly reducing the computational complexity and avoiding overfitting. The caption generation is partly conditioned on textual AudioSet tags extracted by a pre-trained classification model which is fine-tuned to maximize the semantic similarity between AudioSet labels and ground truth captions. To mitigate the data scarcity problem of Automated Audio Captioning (AAC) we pre-train our model on an enlarged dataset. Moreover, we propose a method to apply Mixup augmentation for AAC. Our best model achieves the SPIDEr score of 0.296.

***Index Terms*—** Automated Audio Captioning, transformer encoder-decoder, text conditioning, pre-training, mixup

1. INTRODUCTION

Automated Audio Captioning (AAC) is an intermodal task that aims to generate textual descriptions for a given audio clip. In order to generate meaningful descriptions, a method needs to capture the sound events present in an audio clip and identify the spatial-temporal relationships between them. It has a great practical potential in various applications such as assisting people who are deaf or hard-of-hearing (DHH).

One of the main challenges of AAC is the lack of sufficient data. To address it, many recent approaches utilize pre-trained models such us PANNs and VGGish networks, leading to great improvement in the final performance [2, 3]. Mei et al. [4] use a pre-trained transformer encoder, as transformers have recently shown to outperform CNNs in audio classification tasks [5]. One limitation of transformer encoders is that the complexity of self-attention grows quadratically with respect to the input sequence, making it hard to train on relatively long audio samples, as those appearing in Clotho dataset, i.e. up to 30 sec. To address those issues, we propose a transformer AAC model that utilizes Patchout faSt Spectrogram Transformer (PaSST) [1] as the encoder. The main differences of PaSST encoder to the one proposed by Mei et al. [4] is the current patch extraction method that involves: (1) A convolutional layer that extracts a feature map from the input spectrogram. (2) Decoupled time and frequency positional encoding. (3) *Patchout*, where parts of the transformer's input sequence are dropped during training, encouraging the model to perform with an incomplete sequence.

To solve the word selection indeterminacy problem of AAC, many approaches integrate keyword information either by pre-training the encoder to predict keywords extracted from the ground truth captions [6] or by conditioning the caption generation on input text [7, 8]. In this work, we infer AudioSet class labels from a pre-trained PaSST model and use them as guiding input text. In order to obtain text that is semantically similar to the ground truth captions and functions as a guiding text we fine-tune PaSST on audio-label pairs extracted from Clotho.

Additionally, we propose a method for using Mixup augmentation [9] for AAC.

The remainder of this report is organized as follows: In Section 2 we discuss our proposed system. In Section 3 we present in detail our implementation. In Section 4 we present our results and in Section 5 we conclude this report.

2. SYSTEM DESCRIPTION

The backbone architecture of our proposed model is based on a traditional sequence-to-sequence structure, which consists of a PaSST encoder and a Transformer decoder. The encoder extracts an abstract embedding sequence from the input, then this sequence is fed to the decoder, which outputs a textual description. We propose a multimodal conditioning scheme where the input sequence fed to the encoder consists of both audio and text information. The input text is obtained by a frozen, fine-tuned PaSST model.

2.1. Patch Extraction

Each spectrogram $\mathcal{C}$ is transformed to a sequence of input features $X_{in} \in R^{L \times h}$ through the following pipeling: (1) The spectrogram is passed through a 2-dimensional convolutional layer with kernel size 16, stride 10 and and output dimension d producing feature map $X_m \in R^{d \times F_m \times T_m}$. (2) Positional embeddings are added. Following the implementation of PaSST we add a frequency embedding $e_p^F \in R^{d \times F_m \times 1}$ and a time embedding $e_p^T \in R^{d \times 1 \times T_m}$ to inject positional information in the feature map. (3) *Strctured Patchout* [1] is applied i.e. a number of frequency bins p_f and time frames p_t are randomly deleted from the extracted featuremap X_m resulting in $X_p \in R^{d \times F_p \times T_p}$ where $F_p = F_m - p_f$ and $T_p = T_m - t_f$. (4) Finally X_p is flattened to $X_{in} \in R^{L \times d}$ where $L = F_p \cdot T_p$.

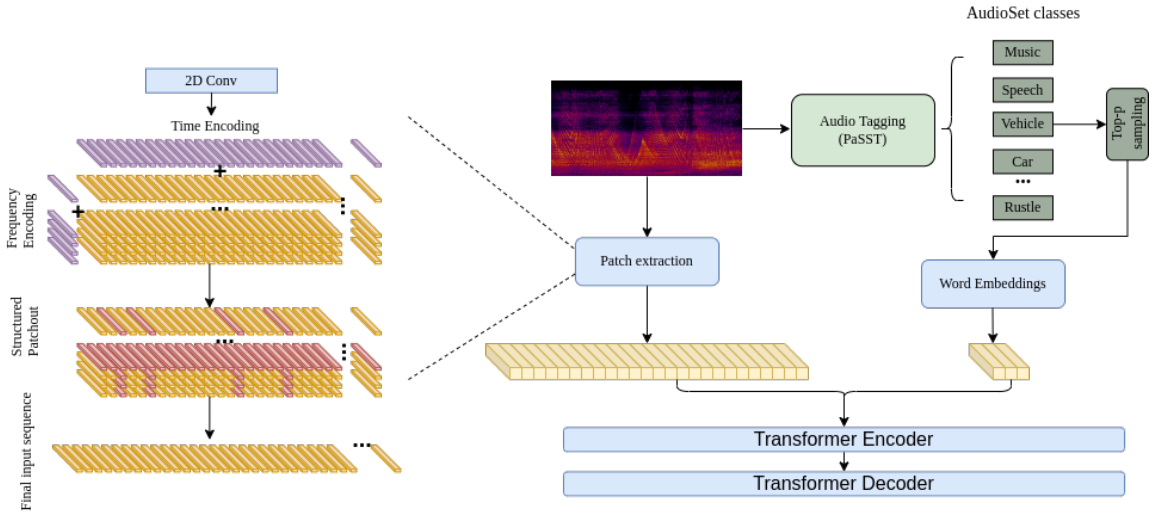

Figure 1: *The overview of our system. Blue and Green colors indicate learned and frozen modules respectively. The input to the decoder is omitted for clarity.*

2.2. Guiding Text

PaSST has recently achieved state-of-the-art performance in audio classification tasks [1]. Using a pre-trained PaSST model, we infer AudioSet class labels from the input audio. Each word in the label is embedded in the input space using trainable embeddings and concatenated with the extracted patches. Similarly to [7], in order to make our system more robust to PaSST's prediction errors, we sample each label from the output distribution. In order to avoid the "unreliable tail" of the distribution we use Nucleus Sampling (top-p) [10]. During inference we select the most probable output label instead of sampling.

Since we add word level information to our model, we want the input label to be semantically similar to the ground truth caption, functioning as a guiding text. We observe that PaSST tends to output labels that capture the general, high level information in the audio and not labels that are more infrequent and specific. Such labels are more likely to be semantically similar or even be present verbatim in the ground truth captions. For example, an audio clip with caption "*A short distance away, a group of people engage in indistinguishable chatter.*" is classified as *Speech* when in fact the AudioSet class *Chatter* would have a higher semantic accuracy. Based on this observation we fine-tune PaSST to predict labels that have the highest sentence similarity with the corresponding ground truth. To achieve this we use a pre-trained BERT model [1] to encode all AudioSet classes and all captions in Clotho dataset. Then we choose the most similar class label for each caption using the cosine similarity of their BERT embedding. We fine-tune PaSST on these audio-label pairs, created from Clotho, by optimizing the standard binary cross entropy loss.

2.3. Encoder

In order to make use of pre-trained models, the encoder architecture is the same as PaSST containing 12 encoder blocks and 12 heads with a hidden dimension $d = 768$. Each layer contains two sub-layers, a multi-head self-attention layer and a position-wise fully-connected feed-forward layer. The feed-forward network contains two linear layers with GELU activation function between them. Since *Patcout* is used to tackle over-fitting we don't use dropout in the transformer encoder. As shown in Figure 1 the input sequence is a concatenation of the extracted patches and the embedded guiding text. As in ViT [11] and AST [5], a global learnable class token $x_{cls} \in R^{1\times d}$ is appended to the beginning of the input sequence. The classification layer of PaSST is omitted and the last hidden layer is passed to the cross-attention layer of the decoder.

2.4. Decoder

The decoder consists of a word embedding layer, a Transformer decoder block, and a linear layer. Each word from the input sequence is embedded through the word embedding layer into a vector $x_i \in R^{512}$ and fed to the transformer decoder. The word vectors are extracted by a pre-trained Word2Vec model [12] trained on the corpus of Clotho dataset.

The transformer decoder contains 6 blocks and 8 heads with an embedding dimension of 512. Each block consists of a self-attention layer, a cross multi-head attention layer and feed-forward layer. The dimension of the forward layer is 2048. The output of the encoder is adjusted through a feed-forward layer and a non-linearity and then fed to the cross multi-head attention layer of the decoder. Positional embeddings are added and masking is applied to the input sequence. Finally, the output of the decoder is passed to a linear layer and a softmax function to get the output probabilities of the caption words.

The training objective of the model is to minimize the cross entropy (CE) loss:

$$\mathcal{L}_{\mathcal{CE}}(\theta) = -\frac{1}{T}\sum_{t-1}^{T} logp(y_t|y_{<t}, \theta) \quad (1)$$

where θ are the models parameters and and y_t is the ground-truth word at time step t.

[1] https://www.sbert.net/

Model	BLUE-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE_L	CIDEr	SPICE	SPIDEr
Baseline	0.555	0.358	0.239	0.156	0.164	0.364	0.358	0.109	0.233
PACT_no_s	0.576	0.384	0.261	0.176	0.166	0.385	0.453	0.130	0.292
PACT_s	0.578	0.384	0.262	0.176	0.177	0.387	0.454	0.133	0.293
PACT_s_clr	0.575	0.384	0.262	0.174	0.178	0.386	0.457	0.133	0.295
PACT_2s	0.579	0.386	0.262	0.173	0.178	0.387	0.457	0.134	0.296

Table 1: Performances of different models on Clotho evaluation split.

2.5. Transfer Learning

Since the use of external data is allowed in this task we experiment with two transfer learning schemes: utilizing a PaSST encoder trained on ImageNet and AudioSet and pre-training our model on a larger in-domain dataset.

2.5.1. Pre-trained encoder

PaSST has an ImageNet pre-trained ViT as its backbone. The model is trained on AudioSet dataset and shows a powerful ability in extracting audio features in different downstream audio pattern recognition tasks [1]. Unlike AST, PaSST uses separate embeddings for time and frequency positional encoding. The advantage of decoupling time and frequency embeddings is allowing variable length inputs without fine-tuning or interpolation.

2.5.2. Pre-training with in domain dataset

We additionally create an enlarged in-domain dataset adding roughly 46,000 single caption audio samples from the AudioCaps dataset [13] and 3930 multiple caption audio samples from MACS [14] to Clotho development-validation split. We first pre-train our model on this dataset and then fine-tune it on Clotho.

2.6. Data Augmentation

In order to avoid over-fitting and regularize the data we use SpecAugment [15], Label Smoothing [16] and Mixup [9]. Since AAC is not a classification task, applying Mixup on the labels is not trivial. Unlike [17] that apply Mixup by concatenating the captions, we mix captions in the embedding space. For two sampled spectrograms x_i, x_j and their corresponding embedded captions y_i^e, y_j^e we apply Mixup as follows:

$$x = \lambda x_i + (1-\lambda)x_j$$
$$y^e = \lambda y_i^e + (1-\lambda)y_j^e$$

where, $\lambda = Beta(a, a)$ and $a = 0.3$. We also mix the embedded guiding text. Despite using unmixed captions when calculating the loss, this implementation of Mixup improves the results of our model.

3. EXPERIMENT

3.1. Dataset

3.1.1. Clotho

The latest Clotho v2 dataset contains 3839 audio clips in the training set and 1045 audio clips in the validation and evaluation set, respectively. The audio clips were collected from Freesound and ranged from 15 to 30 seconds. Annotators were employed through Amazon Mechanical Turk for crowdsourcing the captions. Each audio clip has five corresponding captions ranging from 8 to 20 words long. To prevent biased annotation, only the audio signal was available to the annotators. Clotho splits were created through a stratification process.

3.1.2. AudioCaps

AudioCaps [13] is the largest audio captioning dataset containing around 46000 samples. All audio clips were sourced from AudioSet and are 10 seconds long. The annotation was conducted by crowdworkers through Amazon Mechanical Turk. Audio samples in the training set have one corresponding caption whereas validation and test audio samples have five.

3.1.3. MACS

MACS [14] consists of audio clips from TAU Urban Acoustic Scenes 2019 dataset. It contains 3930, 10 second long audio clips without providing subsplits. The number of captions varies from two to five captions per audio sample.

3.2. Data pre-processing

All audio clips are converted to 32k Hz. Log mel-spectrograms are extracted using a 1024-points Hanning window with 50% overlap and 128 mel bins are used as the input features. Captions are tokenized and transformed to lower case with punctuation removed. <SOS> and <EOS> tokens are are added at the beginning and the end of each caption. During pre-training on the augmented dataset, words that are not present in Clotho vocabulary are replaced with <UNK>.

3.3. Experiment Setup

We split the pretraining stage in to two parts as in [18]. First we freeze PaSST encoder and train on the enlarged dataset with learning rate 1×10^{-4}. Then the encoder is unfrozen and and a learning rate of 1×10^{-5} is used. During fine-tuning an initial learning rate of 1×10^{-5} is gradually increased to 1×10^{-4} using linear warmup. Batch size is 32 throughout all stages. In the inference stage, we adopt beam search with a beam width of 3.

During pre-training we apply *Patchout*, dropping 4 frequency bins and 80 time frames and during fine-tuning on Clotho, 120 time frames. This means that during fine-tuning almost half of the input sequence is dropped resulting to a great complexity reduction. To further mitigate over-fitting we use dropout in the decoder with a rate of 0.2. The label smoothing factor is set to 0.1.

We fine-tune a pre-trained PaSST model on the audio-label pairs we created for 1 epoch with a learning rate of 1×10^{-5}. The input guiding texts for the development-validation splits are obtained prior to training.

4. RESULTS

Our submission contains the following four models:

- PACT_no_s: This model is trained with time Patchout $p_t = 80$. The input text is selected as the maximum value over PaSST's logits.
- PACT_s: This model is trained with time Patchout $p_t = 120$. The input text is selected using top-p sampling.
- PACT_s_clr: Same as PACT_s, fine-tuned on Clotho with a constant learning rate of 1×10^{-5}, instead of linear warm-up.
- PACT_2s: This model is trained with time Patchout $p_t = 120$. The input text is a concatenation of 2 AudioSet class labels selected using top-p sampling.

The performances of our submitted models are compared with the Baseline in Table 1.

5. CONCLUSION

In this report, we describe our system submitted to DCASE2021 challenge, Task 6. We propose a transformer architecture with combined audio and textual conditioning. We show that *Patchout* can be effectively applied for Audio Captioning, reducing the computational complexity and optimizing performance. To obtain input guiding text that is semantically similar to the ground truth captions we fine-tune a pre-trained classification model. To solve the data scarcity problem we pre-train our model on an larger in-domain dataset and initialize the weights of our encoder with a pre-trained PaSST. The SPIDEr score of our best model on the development-testing dataset is 0.296.

6. REFERENCES

[1] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *CoRR*, vol. abs/2110.05069, 2021. [Online]. Available: https://arxiv.org/abs/2110.05069

[2] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. L. Tang, *et al.*, "An encoder-decoder based audio captioning system with transfer and reinforcement learning," *arXiv preprint arXiv:2108.02752*, 2021.

[3] Z. Ye, H. Wang, D. Yang, and Y. Zou, "Improving the performance of automated audio captioning via integrating the acoustic and semantic information," *CoRR*, vol. abs/2110.06100, 2021. [Online]. Available: https://arxiv.org/abs/2110.06100

[4] X. Mei, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Audio captioning transformer," *arXiv preprint arXiv:2107.09817*, 2021.

[5] Y. Gong, Y. Chung, and J. R. Glass, "AST: audio spectrogram transformer," *CoRR*, vol. abs/2104.01778, 2021. [Online]. Available: https://arxiv.org/abs/2104.01778

[6] Z. Ye, H. Wang, D. Yang, and Y. Zou, "Improving the performance of automated audio captioning via integrating the acoustic and semantic information," *arXiv preprint arXiv:2110.06100*, 2021.

[7] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *DCASE 2021 - 6th Workshop on Detection and Classification of Acoustic Scenes and Events*, Virtual, Spain, Nov. 2021. [Online]. Available: https://hal.inria.fr/hal-03522488

[8] Y. Koizumi, Y. Ohishi, D. Niizumi, D. Takeuchi, and M. Yasuda, "Audio captioning using pre-trained large-scale language model guided by audio-based similar caption retrieval," *arXiv preprint arXiv:2012.07331*, 2020.

[9] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[10] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, "The curious case of neural text degeneration," *arXiv preprint arXiv:1904.09751*, 2019.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, 2020. [Online]. Available: https://arxiv.org/abs/2010.11929

[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[13] 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

[14] I. Martin Morato and A. Mesaros, "Diversity and bias in audio captioning datasets," in *Proceedings of the 6th Workshop on Detection and Classication of Acoustic Scenes and Events (DCASE 2021)*, F. Font, A. Mesaros, D. P.W. Ellis, E. Fonseca, M. Fuentes, and B. Elizalde, Eds., Nov. 2021, pp. 90–94, detection and Classication of Acoustic Scenes and Events ; Conference date: 15-11-2021 Through 19-11-2021.

[15] 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," *arXiv preprint arXiv:1904.08779*, 2019.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[17] N. Kuzmin and A. Dyakonov, "Detection and classification of acoustic scenes and events 2020 automated audio captioning technical report."

[18] W. Yuan, Q. Han, D. Liu, X. Li, and Z. Yang, "The DCASE 2021 challenge task 6 system: Automated audio captioning with weakly supervised pre-traing and word selection methods," DCASE2021 Challenge, Tech. Rep., July 2021.