AUDIO CAPTIONING TRANSFORMER

Xinhao Mei[1]*, Xubo Liu*[1]*, Qiushi Huang*[2]*, Mark D. Plumbley*[1]*, Wenwu Wang*[1]

[1] Centre for Vision, Speech and Signal Processing (CVSSP),
[2] Department of Computer Science,
University of Surrey, UK

ABSTRACT

Audio captioning aims to automatically generate a natural language description of an audio clip. Most captioning models follow an encoder-decoder architecture, where the decoder predicts words based on the audio features extracted by the encoder. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are often used as the audio encoder. However, CNNs can be limited in modelling temporal relationships among the time frames in an audio signal, while RNNs can be limited in modelling the long-range dependencies among the time frames. In this paper, we propose an Audio Captioning Transformer (ACT), which is a full Transformer network based on an encoder-decoder architecture and is totally convolution-free. The proposed method has a better ability to model the global information within an audio signal as well as capture temporal relationships between audio events. We evaluate our model on AudioCaps, which is the largest audio captioning dataset publicly available. Our model shows competitive performance compared to other state-of-the-art approaches.

Index Terms— Audio captioning, Transformer, sequence-to-sequence model, cross-modal task

1. INTRODUCTION

Automated audio captioning (AAC) is concerned with describing an audio clip using natural language and is a cross-modal translation task at the intersection of audio processing and natural language processing. Generating a meaningful description for an audio clip not only needs to determine what audio events are presented, but also needs to capture and express their spatial-temporal relationships. Audio captioning is practically useful in applications such as assisting the hearing-impaired to understand environmental sounds, retrieving multimedia content, and analyzing sounds for security surveillance.

Unlike image and video captioning, which have been studied in computer vision (CV) for a longer time, audio captioning is a task investigated only recently [1]. With the announcement of the AAC task in DCASE 2020 and 2021, this topic has attracted increasing attention, and several methods have been proposed [2, 3, 4]. The AAC task is usually treated as a sequence-to-sequence problem, and existing methods are typically based on an encoder-decoder architecture, where the decoder generates words according to the audio features extracted by the encoder. Early works often adopted an "RNN-RNN" architecture with an attention mechanism [1, 3]. However, RNNs can be limited in modeling long-term temporal dependencies in an audio signal. Recently, CNNs have become a dominant approach in audio-related tasks (audio tagging and sound event detection) [5], with many researchers using pre-trained CNNs as the audio encoder, which significantly improved the performance in these systems [6]. More recently, inspired by the great success of the Transformer model in natural language processing [7], the RNN decoder has been replaced by a Transformer decoder in captioning models, and the "CNN+Transformer" architecture has been shown to achieve state-of-the-art performance in this area [8, 9].

Description of an audio signal needs to capture temporal-spatial relationships between audio objects that may be far apart in time. However, convolution is a local operator and has limitations in modelling temporal information, especially with a long audio signal. This can be alleviated by enlarging receptive fields with deeper convolutional layers. However, such deep CNNs can be hard to train and can lead to over-fitting. To address this problem, we propose an Audio Captioning Transformer (ACT), a convolution-free Transformer network based on the self-attention mechanism. We use log mel-spectrograms as input and split the mel-spectrograms into smaller non-overlapping patches along the time axis. By adopting the self-attention mechanism, each patch can attend to all the other patches at each layer of the encoder, which can model global long-range dependencies among the small mel-spectrogram patches from the beginning. Without the need for down-sampling, the features extracted by Transformer are fine-grained, which can contain detailed local audio topics.

The Transformer usually requires more training data than CNNs [10]. However, the amount of data currently available for audio captioning is relatively small. To address this issue, the ACT encoder is firstly pre-trained on AudioSet dataset [11] as an audio tagging task in order to improve its generalization ability. A class token designed to model the global information of an audio clip is appended at the beginning of each patch sequence and is used to output audio tagging results. As a result, when generating words, the decoder can attend to local and global information of an audio clip simultaneously. The proposed ACT model is evaluated on the AudioCaps dataset [3] and shows competitive performance as compared to other state-of-the-art methods.

The remaining sections of this paper are organised as follows. In Section 2, we introduce the related work. The proposed model is described in detail in Section 3. Experimental settings are shown in Section 4. Results are discussed in Section 5. Finally, we conclude our work in Section 6.

2. RELATED WORK

Previous work proposed in audio captioning has been based on deep learning methods with an encoder-decoder architecture. Drossos et al. [1] proposed the first approach to AAC using an RNN-based encoder-decoder architecture with an alignment model in between. To control the information contained in the output text, Ikawa and Kashino [4] introduced a conditional parameter called "specificity" to guide the caption generation. With the release of two freely avail-

able datasets AudioCaps [3] and Clotho [12], AAC has attracted increasing attention and more approaches have been proposed. Kim et al. [3] proposed a model with a top-down multi-scale encoder and aligned semantic attention, which enabled the joint use of multi-level features and semantic attributes. As CNNs have achieved state-of-the-art performance in audio tagging and sound event detection tasks [5], some researchers replaced the RNN encoder with CNNs, which brings significant performance gains [8, 6]. Recently, Transformer has been introduced as the language decoder with a powerful ability in natural language generation tasks [8, 13, 14]. Takeuchi et al. [15] formulated audio captioning as a multi-task learning problem, where they proposed keywords estimation and sentence length estimation to avoid the indeterminacy of word selection. Koizumi et al. [16] utilized a pre-trained large-scale language model GPT-2 [17] with audio-based similar caption retrieval to guide the caption generation. Liu et al. [18] introduced a contrastive loss to get better alignment between audio and texts in the latent space. Reinforcement learning was used to optimize the audio captioning models with non-differentiable evaluation metrics [19].

The Transformer was originally proposed for machine translation and has now become the dominant approach in natural language processing tasks [7]. Recently, many researchers adopted the Transformer for computer vision tasks which was shown to approach or outperform the state-of-the-art CNNs-based systems in image recognition. Dosovitskiy et al. [10] proposed a Vision Transformer (ViT) which was based purely on the attention mechanism, i.e. without using convolution kernels, and applied directly to sequences of image patches for the image classification task. However, a large amount of data are required for pre-training the Transformer models, which limits their adoption. To address this problem, Touvron et al. [20] introduced Data-efficient image Transformers (DeiT) using a data efficiency training and distillation strategy. Based on ViT and DeiT, Liu et al. [21] proposed a CaPtion TransformeR (CPTR) for image captioning. As the Transformer is designed to deal with sequential data, we argue that the Transformer can be adapted for audio signals, and the self-attention mechanism makes it more suitable to capture temporal relationships between audio features and to model the global information. Inspired by these ViT-related works, we propose the Audio Captioning Transformer (ACT) for audio captioning, which, to our knowledge, has not been done in the literature.

3. PROPOSED METHOD

Fig. 1 shows the proposed Audio Captioning Transformer model, which is based on the traditional sequence-to-sequence architecture and is convolution-free. The model takes the log mel-spectrogram of an audio clip as input and outputs the posterior probabilities of the predicted words.

3.1. Encoder

Let $X \in \mathbb{R}^{T \times F}$ denote the log mel-spectrogram of an audio clip, where T is the number of time frames and F is the number of mel bins. The log mel-spectrogram is first split into N non-overlapping small patches $X_N = \{x_1, ..., x_n\}$ along the time axis with size of $t \times F$ where $N = T/t$ and t is the number of time frames of each patch. Then each mel-spectrogam patch is flattened to a 1D embedding and projected to a latent space through a learnable matrix $W_e \in \mathbb{R}^{(t \times F) \times d}$, where d is the dimension of the latent embedding. In line with ViT and DeiT, a global learnable class token $X_{\text{cls}} \in \mathbb{R}^{1 \times d}$ is appended to the beginning of the patch sequences,

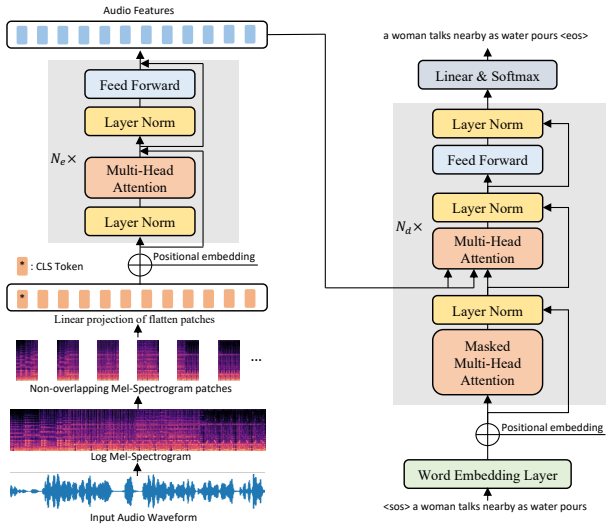

Figure 1: System overview of Audio Captioning Transformer, the encoder is on the left side while the decoder on the right side.

which contains the global information for the audio clip. As the self-attention mechanism cannot capture position information [7], a trainable positional embedding $X_{\text{pos}} \in \mathbb{R}^{(T+1) \times d}$ is added to each patch embedding. Mathematically, the final input representation is given by

$$X_e = [X_{\text{cls}} + W_e X] + X_{\text{pos}} \tag{1}$$

The ACT encoder consists of N_{e} stacked identical layers. Each layer contains two sub-layers, a multi-head self-attention layer and a position-wise fully-connected feed-forward layer. In the self-attention sub-layer, the input is first transformed into query Q, key K and value V through matrix multiplication with three learnable matrices $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$, where d_k is the dimension of each attention head. Then the scaled dot-product attention is computed as

$$\text{Attn}(Q, K, V) = \text{Softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

Each self-attention layer contains h attention heads which extends the model's ability to attend to different positions and creates multiple representation subspaces [7]. The outputs of heads are then aggregated through a linear transformation matrix $W_o \in \mathbb{R}^{(h \times d_k) \times d_k}$, which can be formulated as

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W_o \tag{3}$$

The feed-forward network contains two linear layers with GLEU activation function and dropout applied between them. Layer normalization is applied before each sub-layer and a residual connection is employed around each of them, such that the output of each sub-layer is given by

$$X_{\text{out}} = X_{\text{in}} + \text{Sub\_layer}(\text{LayerNorm}(X_{\text{in}})) \tag{4}$$

In order to make use of pre-trained models, the encoder architecture is the same as ViT and DeiT containing 12 encoder blocks and 12 heads with an embedding dimension of 768.

Model	embedding dim	# layers (N_d)	# heads
ACT_s	512	2	4
ACT_m	512	4	8
ACT_l	512	6	8

Table 1: Variants of the proposed ACT decoder.

3.2. Decoder

The ACT decoder contains three parts: a word embedding layer, a Transformer decoder block, and a linear layer. Each input word is embedded through the word embedding layer into a fixed dimension word vector and then fed into the Transformer decoder block. The word vectors are pre-trained by a Word2Vec model on all caption corpus [22].

The Transformer decoder consists of N_d identical stacked layers. There are two main differences compared to the ACT encoder block. First, the first self-attention sub-layer in the decoder is a masked self-attention because the caption generating process is causal and auto-regressive. Second, there is a new cross multi-head attention sub-layer between self-attention sub-layer and feed-forward sub-layer, which allows every position in the decoder to attend over all positions in the audio features extracted by the encoder [7]. The output of the decoder module is fed through a final linear layer with a softmax activation function to output a probability distribution over the vocabulary.

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

$$\mathcal{L}_{\mathrm{CE}}(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\log p(y_t|y_{1:t-1},\theta) \tag{5}$$

where y_t is the ground-truth word at time step t and θ are the model parameters. The "Teacher forcing" strategy is used during training, i.e. each word to be predicted is conditioned on previous ground-truth words. We experiment with three models, which share the same encoder architecture described in Section 3.2 but have different number of layers and heads in the decoder. Table 1 summarizes the parameters in the decoder of these models.

4. EXPERIMENTS

4.1. Dataset

4.1.1. AudioSet

AudioSet is a large-scale audio dataset with an ontology of 527 sound classes [11]. AudioSet contains more than 2 million 10-second audio clips extracted from YouTube videos. As some audio clips are no longer downloadable, there are 1 934 187 and 18 887 audio clips in our training and evaluation set, respectively. Each audio clip can have one or more labels for their presented audio events.

4.1.2. AudioCaps

AudioCaps is the largest audio captioning dataset currently available with around 50k audio clips sourced from AudioSet [3]. AudioCaps is divided into three splits. Each audio clip in the training set contains one human-annotated caption, while each contains five captions in the validation and test set.

4.2. Data pre-processing

All audio clips in these two datasets are converted to 32k Hz and padded to 10-second long. Log mel-spectrograms extracted using a 1024-points Hanning window with 50% overlap and 64 mel bins are used as the input features. Each log mel-spectrogram is split into 125 non-overlap small patches with the size of 64×4 along the time axis. SpecAugment [23] is applied to augment the input features during training.

Captions are tokenized and transformed to lower case with punctuation removed. To indicate the start and end of each caption, two special tokens "<sos>" and "<eos>" are padded. The vocabulary of AudioCaps contains 5277 distinct words.

4.3. Audio tagging pre-training

As proved in previous works, Transformer requires more training data to achieve competitive performance with CNNs [10]. However, the amount of training data in audio processing area is much less than that in computer vision. Cross-modal transfer learning from ImageNet pre-trained models to audio-related tasks proves to be effective [24]. Thus we make use of pre-trained DeiT models for image classification to initialize the parameters in ACT encoder [10, 20]. As images have three channels and spectrograms just have one channel, we take the average of the weights from the patch embedding layer in DeiT in order to adapt it for spectrogram.

As pre-trained audio neural networks (PANNs) proved to perform well in audio captioning [9], we pre-train ACT encoder on AudioSet as an audio tagging task in order to solve the data scarcity problem and learn more generalized audio patterns. Audio tagging is a multi-classification task of predicting the presence or absence of sound classes within an audio clip [25]. The class token output from the encoder is fed through a linear layer with sigmoid activation function to output the audio events probabilities. The model is trained to minimize the binary cross-entropy loss between the output of the model and the true label

$$\mathcal{L}_{\mathrm{BCE}}(\theta) = -\sum_{n=1}^{N}(y_n \cdot \ln f(x_n) + (1-y_n)\cdot \ln(1-f(x_n))) \tag{6}$$

where x_n is the n-th audio clip in AudioSet and N is the number of training samples. $f(x_n) \in [0,1]^K$ is the output of the model and $y_n \in \{0,1\}^K$ is the true label where K is the number of sound classes. The ACT encoder is pre-trained for 20 epochs with batch size of 128 and learning rate of 1×10^{-4}, which achieves a mean average precision (mAP) of 0.43 on the evaluation set of AudioSet dataset.

4.4. Experimental setups

We train the proposed model for 30 epochs using Adam optimizer [26] and a batch size of 32. The learning rate is linearly increased to 1×10^{-4} in the first five epochs using warm-up, which is then multiplied by 0.1 every 10 epochs. To mitigate over-fitting problem, dropout with rate of 0.2 is applied in the whole model. Label smoothing [27] with a smoothing factor of 0.1 is used to avoid over-confident prediction. We use beam search with a beam size up to 5 to improve the decoding performance during inference stage.

4.5. Evaluation metrics

In line with previous works, we evaluate our methods using machine translation and captioning metrics [13]. BLEU$_n$, ROUGE$_l$

Model	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	METERO	CIDEr	SPICE	SPIDEr
ACT_s_DeiT_AudioSet	0.643	0.483	0.352	0.249	0.469	0.218	0.669	0.160	0.415
ACT_m_DeiT_AudioSet	0.653	**0.495**	**0.363**	**0.259**	**0.471**	0.222	0.663	0.163	0.413
ACT_l_DeiT_AudioSet	0.647	0.488	0.356	0.252	0.468	0.222	0.679	0.160	0.420
ACT_m_scratch	0.567	0.411	0.285	0.191	0.417	0.187	0.501	0.127	0.314
ACT_m_DeiT	0.606	0.445	0.319	0.224	0.445	0.207	0.586	0.147	0.367
RNN+RNN [3]	0.614	0.446	0.317	0.219	0.450	0.203	0.593	0.144	0.369
CNN+RNN [6]	**0.655**	0.476	0.335	0.231	0.467	**0.229**	0.660	**0.168**	0.414
CNN+Transformer [9]	0.641	0.479	0.344	0.236	0.469	0.221	**0.693**	0.159	**0.426**
CNN+Transformer_scratch [9]	0.610	0.461	0.334	0.234	0.455	0.206	0.629	0.144	0.386

Table 2: Scores of the ACT model on the AudioCaps test set. DeiT: the ACT encoder is initialized with the parameters in DeiT, AudioSet: the ACT encoder is pre-trained on AudioSet.

and METEOR are machine translation metrics. $BLEU_n$ is a modified precision metric with a sentence-brevity penalty, calculated as a weighted geometric mean over different length n-grams. $ROUGE_l$ calculates F-measures by counting the longest common subsequence. METEOR evaluates a caption by computing a harmonic mean of precision and recall based on explicit word-to-word matches between the caption and given references. Captioning metrics contain $CIDE_r$, SPICE and $SPIDE_r$. $CIDE_r$ calculates the cosine similarity between term frequency inverse document frequency (TF-IDF) weighted n-grams. SPICE creates scene graphs for captions and calculates F-score based on tuples in the scene graphs. SPIDEr is the average of SPICE and $CIDE_r$ and is selected as the official ranking metric in DCASE challenge, the SPICE score ensures captions are semantically faithful to the audio content, while $CIDE_r$ score ensures captions are syntactically fluent.

5. RESULTS

5.1. Performance comparison

Table 2 presents the results on AudioCaps test set. We compare the proposed ACT model with three representative audio captioning models, "RNN+RNN" [3], "CNN+RNN" [6] and "CNN+Transformer" [9]. In these models, CNNs are all pre-trained on upstream audio-related tasks. As can be seen in Table 2 that the ACT model outperforms "RNN+RNN" model substantially in all evaluation metrics and achieves slightly higher scores than "CNN+RNN" model in most metrics. Compared with the state-of-the-art "CNN+Transformer" approach, ACT model outperforms it in machine translation metrics but gives slightly lower scores in $CIDE_r$. As machine translation metrics are mostly based on n-grams, these results show that the ACT model has better ability in generating words accurately. In addition, training an ACT model is faster than "CNN+Transformer" architecture, where the former takes less than five minutes for one epoch and "CNN+Transformer" needs seven minutes in our experiments. In summary, the ACT model shows competitive performance as compared to other state-of-the-art approaches, and it is simple as it is based only on the self-attention mechanism.

5.2. Ablation studies

The ablation studies are carried out to investigate the effectiveness of the pre-trained encoder and the influence of the hyper-parameters in the decoder. From the experimental results, we can see that pre-training the ACT encoder can boost the performance significantly. Even only using the pre-trained DeiT model, which is originally trained for image classification task, can bring significant performance gains in all the evaluation metrics. Pre-training on AudioSet as an audio tagging task further improves the system to approach the state-of-the-art performance. We also compare the ACT model with the "CNN+Transformer" model both trained from scratch, the results show that the ACT model performs worse than "CNN+Transformer" without encoder pre-training. These results suggest that pre-training the ACT encoder with a large dataset is important, and prove that Transformer network needs more training data than CNNs to achieve competitive performance.

We perform experiments on the three models with different numbers of layers and heads in the decoder. From the observations, the ACT model is slightly sensitive to the choice of hyper-parameters in the decoder. These three models achieve similar performance, among which ACT_m with four decoder layers performs better in machine translation metrics, while ACT_l achieves higher $CIDE_r$ and $SPIDE_r$ scores. The ACT model only needs shallow Transformer decoder layers compared to machine translation models in natural language tasks which typically contain 12 Transformer decoder layers [7]. There might be two reasons. First, the amount of training data in audio captioning is far less than data in natural language processing tasks. Second, the length of the audio captions are usually shorter than sentences in the natural language tasks.

6. CONCLUSION

We have presented a novel audio captioning model, Audio Captioning Transformer (ACT), which is a full Transformer model based on the self-attention mechanism. The encoder of the proposed ACT model can model the global and fine-grained information within an audio signal simultaneously, and has better ability to capture temporal relationships between audio events than CNNs. Experimental results show that the ACT model can outperform other state-of-the-art audio captioning systems in most metrics. Further research should be carried out to adapt the ACT model for audio clips of varied lengths.

7. ACKNOWLEDGMENT

This work is partly supported by grant EP/T019751/1 from the Engineering and Physical Sciences Research Council (EPSRC), a Newton Institutional Links Award from the British Council, titled "Automated Captioning of Image and Audio for Visually and Hearing Impaired" (Grant number 623805725) and a Research Scholarship from the China Scholarship Council (CSC) No. 202006470010.

References

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

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

[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*, 2019, pp. 119–132.

[4] S. Ikawa and K. Kashino, "Neural audio captioning based on conditional sequence-to-sequence model," *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, 2019.

[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," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[6] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 905–909.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[8] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on Transformer and pretrained CNN," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, 2020, pp. 21–25.

[9] 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.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[11] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, New Orleans, LA, 2017.

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

[13] A. Tran, K. Drossos, and T. Virtanen, "WaveTransformer: A novel architecture for audio captioning based on learning temporal and time-frequency information," *arXiv preprint arXiv:2010.11098*, 2020.

[14] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, and S. Saito, "A Transformer-based audio captioning model with keyword estimation," *arXiv preprint arXiv:2007.00222*, 2020.

[15] D. Takeuchi, Y. Koizumi, Y. Ohishi, N. Harada, and K. Kashino, "Effects of word-frequency based pre-and post-processings for audio captioning," *arXiv preprint arXiv:2009.11436*, 2020.

[16] 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.

[17] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[18] X. Liu, Q. Huang, X. Mei, T. Ko, H. L. Tang, M. D. Plumbley, and W. Wang, "Cl4ac: A contrastive loss for audio captioning," *arXiv preprint arXiv:2107.09990*, 2021.

[19] X. Xu, H. Dinkel, M. Wu, and K. Yu, "A CRNN-GRU based reinforcement learning approach to audio captioning," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop*, 2020, pp. 225–229.

[20] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image Transformers & distillation through attention," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 347–10 357.

[21] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full Transformer network for image captioning," *arXiv preprint arXiv:2101.10804*, 2021.

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

[23] 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.

[24] Y. Gong, Y.-A. Chung, and J. Glass, "PSLA: Improving audio event classification with pretraining, sampling, labeling, and aggregation," *arXiv preprint arXiv:2102.01243*, 2021.

[25] Q. Kong, C. Yu, Y. Xu, T. Iqbal, W. Wang, and M. D. Plumbley, "Weakly labelled audioset tagging with attention neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 11, pp. 1791–1802, 2019.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.