INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Rethinking Transfer and Auxiliary Learning for Improving Audio Captioning Transformer

Wooseok Shin[†1], *Hyun Joon Park*[†1], *Jin Sob Kim*[†1], *Dongwon Kim*[2], *Seungjin Lee*[2], *Sung Won Han*[*1]

[1]School of Industrial and Management Engineering, Korea University, Seoul, Republic of Korea
[2]SK Telecom, Republic of Korea

[1]{wsshin95,winddori2002,jinsob,swhan}@korea.ac.kr, [2]{dongwon.kim, joshua.lee}@sk.com

Abstract

The performance of automated audio captioning (AAC) has been improved considerably through a transformer-based encoder and transfer learning. However, their performance improvement is constrained by the following problems: (1) discrepancy in the input patch size between pretraining and fine-tuning steps. (2) lack of local-level relations between inputs and captions. In this paper, we propose a simple transfer learning scheme that maintains input patch sizes, unlike previous methods, to avoid input discrepancies. Furthermore, we propose a patch-wise keyword estimation branch that utilizes an attention pooling method to effectively represent both global- and local-level information. The results on the AudioCaps dataset reveal that the proposed learning scheme and method considerably contribute to performance gain. Finally, the visualization results demonstrate that the proposed attention-pooling method effectively detects local-level information in the AAC system.

Index Terms: audio captioning, transformer, transfer learning, multiple instance learning, attention pooling

1. Introduction

Automated audio captioning (AAC) is the automatic generation of contextual descriptions of audio clips. The AAC system describes the environmental events of audio, instead of the linguistic content, for use in applications such as advanced subtitle generation that is not provided by the script, aiding hearing-impaired people in understanding surrounding sounds, and automatic content summarization.

Typically in AAC, a sequence-to-sequence framework is used, that comprises an encoder extracting acoustic features from the audio input and a decoder generating captions using the extracted features. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) have been widely adopted as encoder architectures [1, 2, 3, 4, 5]. Moreover, RNN-based architectures have been used as decoder architectures [1, 2, 5]. Recently, transformers have been proven to outperform existing architectures in natural language processing (NLP), computer vision, and speech [6, 7, 8]. Therefore, the adoption of a transformer for AAC has attracted considerable research attention [3, 4, 9]. Mei *et al.* [9] proposed a full transformer structure called an audio captioning transformer (ACT) that achieved state-of-the-art performance on the AudioCaps dataset [10]. However, the performance gain of ACT is still limited owing to the following two problems: discrepancy in the patch size and lack of relations between inputs and captions.

First, as AAC tasks have limited labeled data, it has become a common approach to pretrain the encoder, including the CNN and transformer, on audio tagging tasks. In particular, as the transformer is generally known to require more training data than the CNN [7], pretraining and transfer learning approaches are essential for obtaining general audio representations. Therefore, in the ACT encoder based on the vision transformer (ViT [7]) architecture, the parameters are initialized using the weights of data-efficient image transformers (DeiT [11]) pretrained on ImageNet. Subsequently, the ACT is trained sequentially on audio tagging and AAC tasks. ACT demonstrates the potential of the pure transformer structure with pretraining; however, ACT does not achieve significant performance improvements compared to the combination of CNN and transformer structures. This is attributed to the input discrepancy between the pretraining and fine-tuning stages. To be specific, DeiT uses 16×16 patch sizes during training, whereas ACT uses patches of 64×4 sizes split along the time axis of the spectrogram. Although this split scheme can preserve frequency-wise information, it may induce discrepancies that impede performance improvement during the fine-tuning step due to differences in the learned relationships between patches.

Second, the learning process of AAC is generally performed between an audio clip and the entire caption label, and between input patches and the entire caption label in the case of ACT. The entire caption label operates as a global-level label which consists of successive local-level sound event descriptions. This hierarchy indicates that the components of the global-level labels can operate as local-level labels. Since the general learning process of AAC predicts only global-level labels, the ability of the model to capture the relationships between inputs and local-level labels is limited, leading to suboptimal performance. One method for strengthening the relationship with local-level labels is to add a keyword estimation branch to the AAC framework [12, 13, 14, 15, 16]. Existing studies extracted nouns and verb keywords, which are components of sound event descriptions, from the entire caption. By predicting keywords as an auxiliary task, the model can learn the relationship between inputs and local-level information. Among existing studies, some studies [12, 13, 15] have revealed that estimating keywords aids a network to generate captions, whereas others [14, 16] have revealed that it may not work universally under varying conditions. This limited improvement may occur owing to the absence of keyword ground truth matching with frame-level inputs. To address this problem, existing methods predict frame-wise keywords and then aggregate them along the time axis by using max or average pooling. However, their aggregating methods are inappropriate for representing the relations with local-level labels due to the following limitations: (1) Max pooling can result in a penalty of only one frame in a clip, potentially ignoring the contribution of other frames. (2) Average pooling can ignore events that have relatively short durations compared to the duration of the clip

[†]These authors contributed equally to this work.
[*]Corresponding author.

10.21437/Interspeech.2023-619

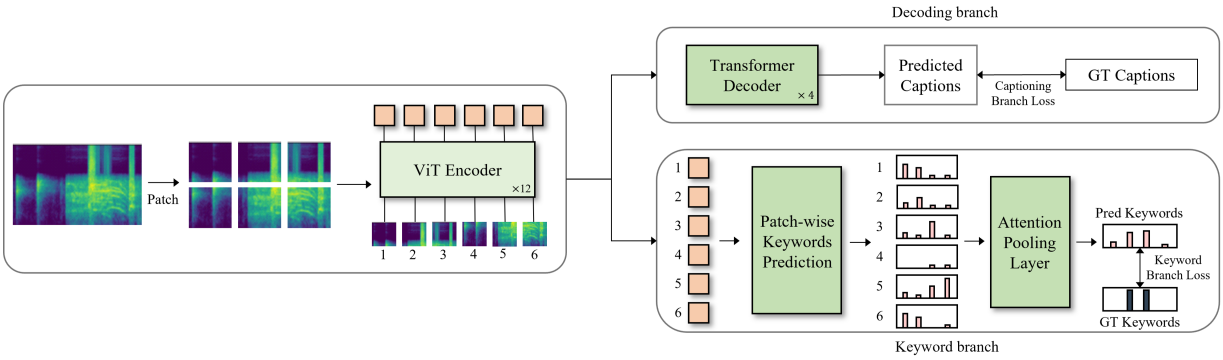

Figure 1: *The overall framework of the proposed method.*

[17, 18, 19].

Based on the investigation of the two aforementioned key problems in AAC, we propose a training approach that maintains the input patch size (16 × 16) from the pretraining to the fine-tuning stages, thereby maximizing the advantage of the pretraining stage. In addition, we design a patch-wise keyword estimation method to enhance the ability of the model to capture the relationship between input patches and keywords representing local-level information. To aggregate patch- to clip-level keyword estimations without biased information, we utilize an attention-pooling method. Experimental results on the AudioCaps dataset reveal that the proposed methods significantly improve performance compared to that of the baseline. Finally, through visualization, we verify how well the proposed keyword estimation method can represent local-level information compared to other methods.

2. Method

2.1. Acoustic Encoder

Given the acoustic signal, the AAC system encoder takes the mel-spectrogram of that as the input. The encoder aims to provide subsequent branches with a hidden representation that encodes information entailed by the acoustic context. We use AST [8] as the architecture of the encoder, which intuitively inherits the ViT approach [7] in the acoustic data processing domain. As depicted in Figure 1, the input mel-spectrogram is first split into a sequence of 16 × 16 patches. The patches are linearly projected onto embedded representations and then processed by transformer encoder blocks [6]. Each transformer block comprises a multi-headed self-attention layer and a feed-forward layer, and both are followed by normalization and residual connection.

To take advantage of transfer learning, similar to priors [3, 4, 9], we exploit an acoustic encoder pretrained on AudioSet [20] with an audio tagging task. Among ViT-oriented acoustic encoders, the pretrained weight of PaSST [21] is used to initiate transformer blocks. In addition, to improve the training efficiency and boost the robustness of the network, Patchout [21] scheme is applied during the training phase.

2.2. Linguistic Decoder

The decoder is responsible for generating captions from information extracted by the acoustic encoder. Taking the output from the encoder, it predicts a sequence of linguistic tokens depicting the circumstances corresponding to the acoustic signal input. We use a typical transformer decoder proposed by [6], as the architecture has been successful in several NLP domains, including AAC [3, 4, 9].

Given the input audio mel-spectrogram $x \in \mathbb{R}^{t \times f}$, which comprises f frequency features for t frames, along with the encoder (f_{enc}) and decoder (f_{dec}), we formulate the autoregressive estimation of the decoding branch as follows:

$$\hat{y}_n = f_{dec}(\hat{y}_0, ..., \hat{y}_{n-1}, f_{enc}(x)), \tag{1}$$

where $\hat{y}_n$ is a probability distribution over vocabulary, fed through a softmax activation function, except $\hat{y}_0 = <\text{sos}>$ the special token to initiate the decoder to generate tokens. Therefore, when the ground truth caption comprises a sequence of M tokens, $[y_1, ..., y_M]$, the training objective of the decoding branch is to minimize the cross-entropy loss, as follows:

$$L_{standard} = -\frac{1}{M}\sum_{m=1}^{M} \log p(y_m|\hat{y}_m) \tag{2}$$

We use the teacher forcing strategy during model training, which is to condition the ground truth $[y_1, ..., y_{m-1}]$ for $\hat{y}_m$ decoder prediction.

2.3. Keyword Estimation

To strengthen the ability of the model to capture local-level information from inputs, we add a keyword estimation branch to the AAC system, as displayed in Figure 1. As an absence of keyword ground truth matching with frame-level or patch-level inputs, we adopt multiple instance learning (MIL) schemes. MIL, which is a useful approach for situations in which only a bag-level label is available for a bag-level input comprising multiple instances, has been widely used in various tasks, such as image classification [22, 23], sentiment analysis [24], medical diagnosis [23, 25], and audio tagging [17, 26].

To incorporate MIL into our framework, we formulate an effective embedding-level MIL approach, as follows. Given a bag-level input (mel-spectrogram x) comprising N instances (patches), N patch-level representations $[x_1, x_2, ..., x_N]$ are extracted by the encoder and passed to the keyword branch. In the keyword branch, a simple neural network f_ψ consisting of a linear layer, layer normalization, and ReLU transforms x_n into a low-dimensional embedding h_n as follows:

$$h_n = f_\psi(x_n) \in \mathbb{R}^D, \tag{3}$$

where D denotes hidden dimensions. Then, to obtain a bag-level representation z, we utilize attention-based MIL pooling [22],

Table 1: *Evaluation results on the AudioCaps test dataset. † indicates that the Patchout training technique is employed for cost-efficient training. TF denotes transformer decoder. For all metrics, higher scores are better. All metrics are reported in percentile (%) values.*

Model	Params (M)	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	METEOR	$ROUGE_L$	CIDEr	SPICE	SPIDEr
Transfer Learning										
AST (128×2) + TF (*from scratch*)	104.6	59.2	42.5	30.0	20.9	18.3	42.0	45.6	13.0	29.3
AST (16×16) + TF (*from scratch*)	104.3	58.9	41.6	29.3	20.2	17.8	41.4	41.2	12.0	26.6
AST (128×2) + TF (*from pretrained*)	104.6	69.5	**52.9**	**38.7**	**27.6**	23.0	48.4	68.1	16.8	42.5
AST (16×16) + TF (*from pretrained*)	104.3	**70.1**	**52.9**	38.5	27.2	**23.5**	**49.7**	**71.8**	**17.2**	**44.5**
Keyword Branch										
†AST (16×16) + TF (w/o Keyword)	104.3	70.1	54.0	**40.2**	**28.8**	23.4	**50.5**	71.9	17.6	44.7
+ Max Pooling	104.9	70.3	53.4	39.3	28.3	23.7	50.1	73.1	17.8	45.4
+ Average Pooling	104.9	70.6	53.7	39.6	28.4	23.9	49.9	72.9	17.9	45.4
+ Attention Pooling (ours)	105.0	**71.6**	**54.4**	39.9	28.5	**24.2**	50.4	**76.4**	**18.0**	**47.2**

which enables adaptive aggregation according to the inputs, as follows:

$$z = \sum_{n=1}^{N} \alpha_n h_n \tag{4}$$

Here, the attention weight α_n, that is, the importance of each patch on the keywords, is measured as follows:

$$\alpha_n = \frac{\exp(\mathbf{W}^T \tanh(\mathbf{U}\, h_n^T))}{\sum_{n=1}^{N} \exp(\mathbf{W}^T \tanh(\mathbf{U}\, h_n^T))}, \tag{5}$$

where $\mathbf{W} \in \mathbb{R}^{L\times 1}$ and $\mathbf{U} \in \mathbb{R}^{L\times D}$ are trainable parameters. Finally, the predicted keyword class probability $\hat{y}$ is obtained by applying a linear layer with a number of nodes equal to the number of pre-defined keywords (K) to the clip-level representation z. As keyword prediction is a multi-label classification problem, the objective function is defined as follows:

$$L_{keyword} = -\sum_{j=1}^{K} y_j \cdot log\, \hat{y_j} + (1 - y_j) \cdot log(1 - \hat{y_j}) \tag{6}$$

The total loss is defined by combining the standard and keyword branch losses as follows:

$$L_{total} = L_{standard} + \beta \times L_{keyword}, \tag{7}$$

where β denotes the keyword branch weight. In this way, the proposed method aids the network to capture not only global-level information but also local-level sound event information.

3. Experiments

3.1. Experimental Configuration

Datasets The AudioCaps dataset [10], the largest audio captioning dataset including approximately 50k audio samples obtained from AudioSet [20] and human-annotated descriptions, is used for validation. The AudioCaps dataset is divided into training, validation, and testing datasets. The training set consists of 49,274 audio clips with one caption per clip, and the validation and test sets consist of 494 and 957 audio clips with five captions per clip, respectively.

Evaluation metrics To evaluate the proposed method, we adopt machine translation and captioning metrics, which are widely used in AAC. For machine translation metrics, $BLUE_N$ [27] is a modified form of n-gram precision that integrates a brevity penalty to evaluate prediction accuracy. It does so by measuring the degree of n consecutive word matches between predicted and reference captions. $ROUGE_L$ [28] computes an F-measure by identifying the longest continuous subsequence in both sentences. METEOR [29] calculates an F-measure by considering several factors, such as stem- and word-level overlap, synonyms, and unigram precision. For captioning metrics, CIDEr [30] considers the semantic similarity between a set of reference captions and a candidate caption by computing the geometric mean between the n-gram and cosine similarity scores. SPICE [31] uses semantic scene graphs obtained from captions to validate the caption quality based on their semantic content. SPIDEr [32], which is used to measure the official ranking in the DCASE challenge is the average of SPICE and CIDEr scores.

Implementation details We use an epoch of 30 and a batch size of 8. We adopt an Adam optimizer and its learning rate is 1×10^{-5}. We use 12 layers, 12 heads, and a hidden size of 768 for the acoustic encoder. The linguistic decoder has 4 layers, 512 hidden dimensions, and 8 heads. For the keyword estimation branch, the hidden size is 512. As training strategies, we exploit the label smoothing with a ratio of 0.1, spectrogram-based augmentation [33], and structured patchout with time frames and frequency bins of 40 and 4 respectively. Furthermore, we extract the input features, the mel-spectrogram, based on a sample rate of 32,000, window size of 25 ms, hop size of 10 ms, and mel bins of 128. Regarding the caption preprocess, we take the same tokenization process as [9], and the process makes the vocabulary with 5,277 unique words. Based on this vocabulary, we use the natural language toolkit (NLTK) [34] to obtain keywords from captions. We filter only the noun-related phrases and create a keyword vocabulary with the top $K = 500$ keywords based on their frequency in the dataset. The loss function weight β is set to 5. For K and β, the optimal values were determined through empirical experiments and used in this study. Finally, captions are generated using a beam search size of 3.

3.2. Results

Table 1 presents the experimental results of the proposed method on the AudioCaps dataset. Since the AST exhibits remarkable performance as an encoder in audio classification tasks, we adopt it as the baseline encoder structure instead of the encoder proposed in ACT [9].

First, to verify the effect of aforementioned discrepancy, we train the model using different patch sizes with the same area size during the AAC training step. As listed in the first block of Table 1, when starting from scratch, the model trained using a patch size split along the time axis from the spectrogram (i.e., 128×2) outperforms the model trained using a size of 16×16. In contrast, when starting from pretrained weights, the model trained using the same patch size (i.e., 16×16) as in the pretraining step outperforms the model trained using a different patch

Table 2: *Comparison with existing state-of-the-art methods on the AudioCaps test dataset.*

Model	Params (M)	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	METEOR	$ROUGE_L$	CIDEr	SPICE	SPIDEr
CNN + TF [4]	-	64.1	47.9	34.4	23.6	22.1	46.9	69.3	15.9	42.6
ACT-M + TF [9]	108	65.3	49.5	36.3	25.9	22.2	47.1	66.3	16.3	41.3
BART + YAMNet + PANNs [35]	494	69.9	52.3	38.0	26.6	24.1	49.3	75.3	17.6	46.5
AL-MixGen [36]	108	69.3	52.9	38.9	28.3	24.1	49.9	75.5	17.7	46.6
†AST + TF + Attention MIL (ours)	105	**71.6**	**54.4**	**39.9**	**28.5**	**24.2**	**50.4**	**76.4**	**18.0**	**47.2**

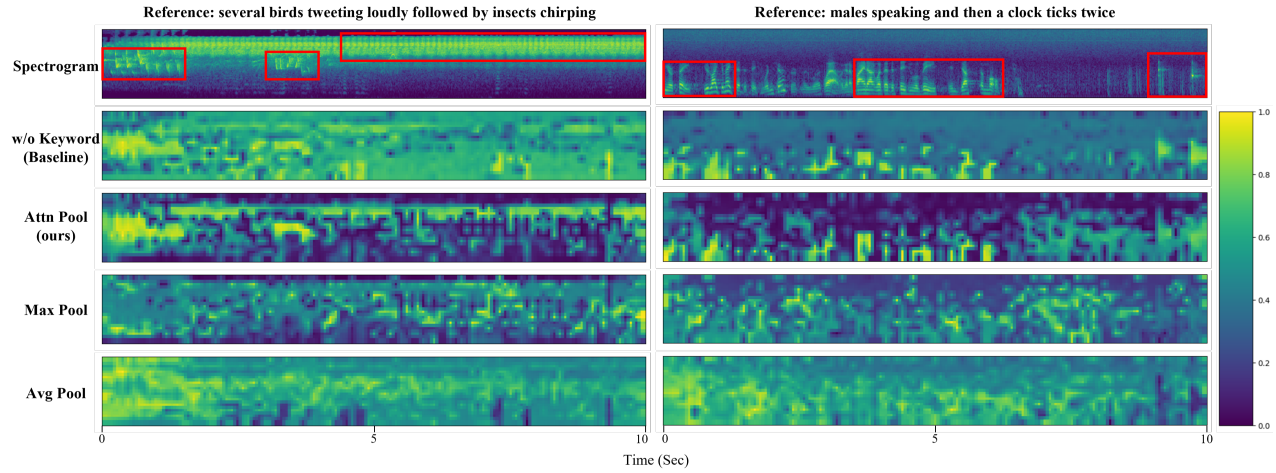

Figure 2: *Heatmap of magnitude (l_2 norm) of the patch representations extracted through the encoder. We randomly selected two samples from the test set and reconstructed sequence patch representations into a spatial spectrogram for visualization.*

size (i.e., 128×2). The results suggest that although preserving the frequency-axis information can be crucial, it does not fully leverage the benefits offered by pretrained knowledge. Alternatively, pretraining 128×2 patch-based models on ImageNet and AudioSet is possible but is computationally expensive. Thus, using 16×16 patches is the most practical and optimal solution for AAC training.

Next, we validate the efficacy of the proposed attention pooling method for keyword estimation in AAC. As reported in the second block of Table 1, conventional pooling methods exhibit comparable performance compared to the baseline without a keyword estimation branch, whereas the proposed method achieves significant improvement. This suggests that the efficacy of existing pooling methods is constrained, whereas attention-based pooling provides proper information that benefits AAC systems by adequately detecting local-level events.

Further, we compare the proposed method with four state-of-the-art methods [4, 9, 35, 36]. The results are presented in Table 2. The model trained using the proposed method significantly outperforms state-of-the-art models in terms of all metrics with fewer model parameters, achieving a new state-of-the-art performance on the AudioCaps dataset.

3.3. Discussion

We attempt to explain the effectiveness of the proposed attention-based pooling method in terms of the magnitude (l_2 norm) of the patch representations. As depicted in Figure 2, existing pooling methods (i.e., max and average) cannot accurately detect the region of sound events. In contrast, the baseline and proposed method could adequately detect the main sound events. These results suggest that existing pooling methods are inadequate for using local-level information effectively. Regarding false positives, the baseline model can adequately capture primary sound events but exhibits a tendency to produce false positives in areas without any sound events, whereas the proposed method is effective in accurately detecting sound event regions without considerable false positives. Finally, the results suggest that the proposed attention-based pooling is a suitable aggregation method for the keyword estimation branch, which aids a network to capture local-level information in the AAC system.

4. Conclusions

In this work, we proposed two strategies to improve the performance of transformer-based networks in AAC: (1) Preventing discrepancies resulting from the difference in input patch size between the pretraining and fine-tuning steps. (2) Suggesting a patch-wise keyword estimation branch that utilizes attention-based pooling to adequately detect local-level information. Experimental results on the AudioCaps dataset indicate that the proposed methods significantly improve the performance compared to the baseline. Furthermore, the model trained using the proposed methods outperforms existing state-of-the-art methods. Finally, we visually verified the effectiveness of the proposed keyword estimation pooling method. The results reveal that the proposed method effectively detects local-level information with minimal false positives compared to other methods.

5. Acknowledgements

This research was supported by Brain Korea 21 FOUR.

6. References

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

[2] S. Ikawa and K. Kashino, "Neural audio captioning based on conditional sequence-to-sequence model," 2019.

[3] K. Chen, Y. Wu, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pre-trained cnn." in *DCASE*, 2020, pp. 21–25.

[4] 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.

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

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] 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.

[8] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

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

[10] 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)*, 2019, pp. 119–132.

[11] 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.

[12] 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.

[13] E. Çakır, K. Drossos, and T. Virtanen, "Multi-task regularization based on infrequent classes for audio captioning," *arXiv preprint arXiv:2007.04660*, 2020.

[14] 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.

[15] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Automated audio captioning with keywords guidance," Tech. rep., DCASE2022 Challenge, Tech. Rep., 2022.

[16] 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.

[17] S. Hong, Y. Zou, and W. Wang, "Gated multi-head attention pooling for weakly labelled audio tagging," in *Interspeech 2020*, 2020.

[18] H. Wang, Y. Zou, and W. Wang, "A global-local attention framework for weakly labelled audio tagging," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 351–355.

[19] X. Wang, X. Zhang, Y. Zi, and S. Xiong, "A frame loss of multiple instance learning for weakly supervised sound event detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 331–335.

[20] J. F. Gemmeke, D. P. 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 *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[21] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, "Efficient training of audio transformers with patchout," *arXiv preprint arXiv:2110.05069*, 2021.

[22] M. Ilse, J. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," in *International conference on machine learning*. PMLR, 2018, pp. 2127–2136.

[23] Z. Shao, H. Bian, Y. Chen, Y. Wang, J. Zhang, X. Ji *et al.*, "Transmil: Transformer based correlated multiple instance learning for whole slide image classification," *Advances in neural information processing systems*, vol. 34, pp. 2136–2147, 2021.

[24] S. Angelidis and M. Lapata, "Multiple instance learning networks for fine-grained sentiment analysis," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 17–31, 2018.

[25] Z. Qian, K. Li, M. Lai, E. I.-C. Chang, B. Wei, Y. Fan, and Y. Xu, "Transformer based multiple instance learning for weakly supervised histopathology image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part II*. Springer, 2022, pp. 160–170.

[26] S.-Y. Tseng, J. Li, Y. Wang, J. Szurley, F. Metze, and S. Das, "Multiple instance deep learning for weakly supervised small-footprint audio event detection," *arXiv preprint arXiv:1712.09673*, 2017.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[28] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[29] S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[30] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[31] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*. Springer, 2016, pp. 382–398.

[32] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 873–881.

[33] 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.

[34] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[35] F. Gontier, R. Serizel, and C. Cerisara, "Automated audio captioning by fine-tuning bart with audioset tags," in *Detection and Classification of Acoustic Scenes and Events-DCASE 2021*, 2021.

[36] E. Kim, J. Kim, Y. Oh, K. Kim, M. Park, J. Sim, J. Lee, and K. Lee, "Improving audio-language learning with mixgen and multi-level test-time augmentation," *arXiv preprint arXiv:2210.17143*, 2022.