Graph Attention for Automated Audio Captioning

Feiyang Xiao, Jian Guan, *Member, IEEE*, Qiaoxi Zhu, *Member, IEEE*,
and Wenwu Wang, *Senior Member, IEEE*

***Abstract*—State-of-the-art audio captioning methods typically use the encoder-decoder structure with pretrained audio neural networks (PANNs) as encoders for feature extraction. However, the convolution operation used in PANNs is limited in capturing the long-time dependencies within an audio signal, thereby leading to potential performance degradation in audio captioning. This letter presents a novel method using graph attention (GraphAC) for encoder-decoder based audio captioning. In the encoder, a graph attention module is introduced after the PANNs to learn contextual association (i.e. the dependency among the audio features over different time frames) through an adjacency graph, and a top-k mask is used to mitigate the interference from noisy nodes. The learnt contextual association leads to a more effective feature representation with feature node aggregation. As a result, the decoder can predict important semantic information about the acoustic scene and events based on the contextual associations learned from the audio signal. Experimental results show that GraphAC outperforms the state-of-the-art methods with PANNs as the encoders, thanks to the incorporation of the graph attention module into the encoder for capturing the long-time dependencies within the audio signal. The source code is available at https://github.com/LittleFlyingSheep/GraphAC.**

***Index Terms*—Audio modelling, temporal information, automated audio captioning, graph attention network.**

I. Introduction

AUTOMATED audio captioning (AAC) aims to describe an audio signal with captions using natural language and focus on non-speech content, such as environmental sound [1]. It can facilitate man-machine interaction for those with hearing loss, sound analysis for security surveillance [2], and automatic content summarisation, e.g., subtitling for the sound of a television program [2], [3].

The encoder-decoder structure is popular for AAC. The audio encoder extracts the audio feature, and the text decoder generates the caption from the audio feature. In early methods [4]–[6], recurrent neural networks (RNNs) [7] and Transformer [8] have been used for audio captioning. However, the encoders used in these methods may not be effective in feature representation, due to the use of either a simple model or the limited amount of training data. As a solution, the pretrained audio neural networks (PANNs) [10] was applied widely as the audio encoder in recent research. The PANNs model is pretrained on a large-scale audio dataset (i.e., AudioSet [11]). This has led to significant performance improvement in audio captioning [2], [3], [9], [12]–[14].

Despite its excellent performance, the convolution operation in PANNs primarily captures information from the local receptive field (i.e., local time-frequency region), ignoring the contextual associations among audio features and their long-time dependencies [15], [16]. Nevertheless, audio signals are time-variant and contain rich temporal information, including long-time dependencies that carry semantic information about the acoustic scene and events. Missing such information may affect the effectiveness of audio representation in the audio encoder and limit the captioning performance.

This letter presents a novel audio captioning (AC) method, namely GraphAC, with a graph attention module incorporated into the encoder for feature representation. Specifically, in GraphAC, P-Transformer [15] is used as the backbone, and the graph attention module is introduced after the PANNs in the audio encoder to obtain more effective audio representation. The graph attention module not only captures the temporal contextual information within the audio signal, i.e., by exploiting the contextual association between audio nodes (i.e., audio feature frames) obtained in the learnt adjacency graph with a top-k mask, but also highlights the important semantic information about the acoustic scene and events in the feature representation, i.e., by aggregating audio nodes with the learnt adjacency graph. As a result, the encoder of the proposed GraphAC acquires a better audio feature due to the exploitation of the contextual information from the longer time duration. This information can improve the accuracy of captions generated by the text decoder (i.e., a Transformer-based decoder).

The graph attention module learns the edge connections between audio feature nodes via the attention mechanism [19], and differs significantly from the graph convolutional network (GCN), which is popular for image and video captioning [17], [18], but uses convolution as the fundamental operation for feature representation. In contrast to GCN for image and video captioning, our GraphAC does not require a pre-trained graph model to generate a graph structure. In addition, it introduces a top-k mask strategy to remove noisy nodes caused by non-zero weights assigned to audio frames in encoder learning [15]. Compared to RNNs and Transformers, as used in official baselines of DCASE Challenge Task 6, which can also model long-time dependencies, our method offers an additional advantage in attending important audio feature nodes and ignoring meaningless audio feature nodes, thereby highlighting the important semantic information about acoustic scenes and events.

Experiments are performed on the DCASE 2021 Challenge

This work was partly supported by the Natural Science Foundation of Heilongjiang Province under Grant No. YQ2020F010, a Newton Institutional Links Award from the British Council with Grant No. 623805725, and a GHfund B with Grant No. 202302026860. (Corresponding author: Jian Guan)

F. Xiao and J. Guan are with the Group of Intelligent Signal Processing (GISP), College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China (emails: xiaofeiyang128@gmail.com; j.guan@hrbeu.edu.cn).

Q. Zhu is with the Centre for Audio, Acoustics and Vibration, University of Technology Sydney, Ultimo, NSW, Australia (email: qiaoxi.zhu@gmail.com).

W. Wang is with the Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, U.K. (email: w.wang@surrey.ac.uk).

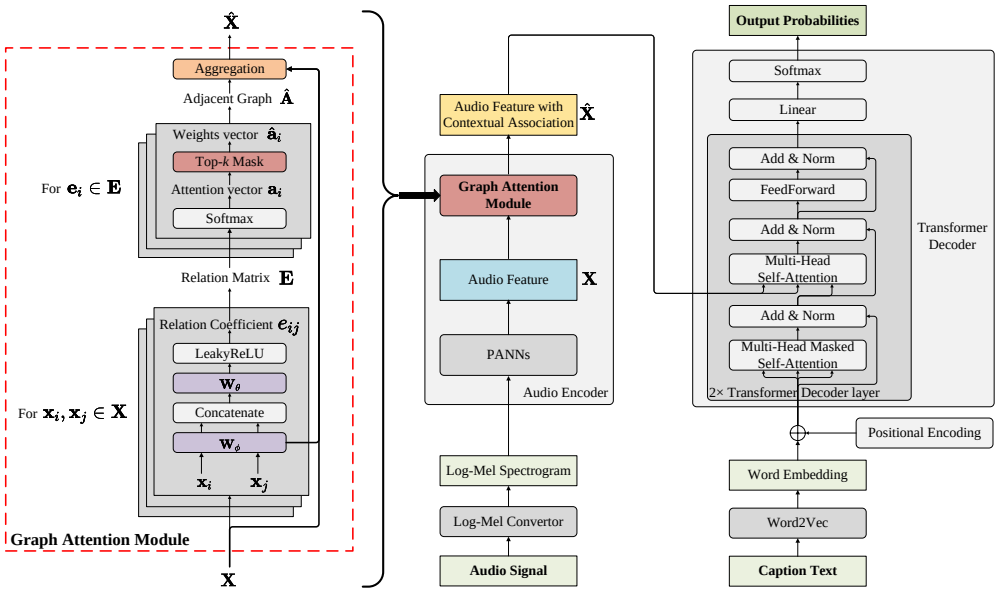


Fig. 1. The framework of our proposed GraphAC method, where P-Transformer [2] is used as the backbone. The difference between GraphAC and P-Transformer is that GraphAC has the graph attention module in the encoder, as shown in the red dashed box.

Task 6 dataset [4], and show that the proposed GraphAC outperforms state-of-the-art techniques in all the assessment criteria by exploiting time dependencies. With the graph attention based audio feature representation, the proposed method can capture important semantic information about the acoustic scene and events which helps improve the captioning performance.

II. Graph Attention Audio Captioning

The proposed GraphAC adopts P-Transformer [2] as the backbone, with the graph attention module added in the encoder to model the relations among the audio nodes (i.e., audio feature frames) extracted by PANNs, and to obtain an improved audio feature representation. Then, a Transformer decoder is employed to predict the caption from the audio feature representation. Fig. 1 shows the framework of GraphAC.

A. Audio Feature Extraction

The PANNs module [10] is applied to extract the audio feature from an audio signal. The audio signal is converted to the log-Mel spectrogram $\mathbf{X}_{Mel}$ as input to the PANNs module (i.e., CNN10). Different from the original CNN10 structure in [10], here, only the global average pooling is used on the Mel-band dimension after the convolutional blocks, and the channel dimension is taken as the audio feature dimension in this work. Then, the dimension of the output of the last two layers is modified to obtain the audio feature $\mathbf{X} \in \mathbb{R}^{T\times D}$, where T denotes the temporal dimension and D denotes the audio feature dimension. Here, D is set empirically as 128.

B. Graph Attention for Audio Feature Representation

Instead of directly using the audio feature extracted by PANNs, a graph attention module is introduced to represent the timing information of the audio signal. The adjacency graph is built in the encoder to represent the audio feature with node relations by graph attention mechanism, where a top-k mask strategy is introduced to remove noisy nodes for better feature representation. Then, the audio feature nodes are aggregated with the learnt contextual association.

*1) **Audio Feature Graph Modelling**:* The contextual relation between audio feature nodes is built by the adjacency graph to exploit the long-time dependencies within the audio feature. The audio feature $\mathbf{X}=[\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T]^\top$ is a set of T audio feature nodes $\mathbf{x}_i \in \mathbb{R}^D$ $(1 \le i \le T)$ and $\top$ denotes the matrix transposition. Each node represents the audio feature over a time frame. The coefficient characterising the relation between two audio feature nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ $(1 \le i, j \le T)$ is calculated by a learnable linear mapping operation via the additive score function following [19]:

$$e_{ij} = LeakyReLU(\mathbf{W}_\theta[\mathbf{W}_\phi\mathbf{x}_i; \mathbf{W}_\phi\mathbf{x}_j]), \tag{1}$$

where $\mathbf{W}_\phi \in \mathbb{R}^{D\times D}$ and $\mathbf{W}_\theta \in \mathbb{R}^{1\times 2D}$ are two matrices containing learnable parameters, and $[\cdot;\cdot]$ denotes the concatenation operation. The matrix $\mathbf{W}_\phi$ maps each audio feature node into a relation embedding space, and the matrix $\mathbf{W}_\theta$ maps the concatenated relation embedding into the relation coefficient e_{ij}. The leaky ReLU is used as the activation function. The relation matrix which is denoted as $\mathbf{E} \in \mathbb{R}^{T\times T}$ with e_{ij} being its element at the i-th row and j-th column, contains the relation coefficients of all pairs of nodes.

Then, we employ the relation matrix to calculate the adjacency graph. The row vector of the relation matrix $\mathbf{e}_i$ $(1 \le i \le T)$ is normalised by the softmax function, resulting in the attention vector $\mathbf{a}_i$. Here, $\mathbf{a}_i$ contains the edge weights between the node $\mathbf{x}_i$ and all nodes, including itself. Then, we adopt a top-k mask strategy for node selection as follows

$$\hat{a}_{ij} = \begin{cases} a_{ij}, & a_{ij} \in \text{top}_k(\mathbf{a}_i) \\ 0, & a_{ij} \notin \text{top}_k(\mathbf{a}_i) \end{cases}, \tag{2}$$

where a_{ij} denotes the input weight between the node $\mathbf{x}_i$ and the node $\mathbf{x}_j$, and $\text{top}_k(\mathbf{a}_i)$ denotes the set of k largest elements

in $\mathbf{a}_i$. With the top-k mask, we can prioritise important relations between audio feature nodes and select k most relevant nodes, while mitigating the interferences from noisy nodes. Finally, the adjacency graph $\hat{\mathbf{A}} \in \mathbb{R}^{T\times T}$ is formed with $\hat{a}_{ij}$ being its ij-th element.

*2) **Graph Nodes Aggregation**:* With the learnt adjacency graph, we aggregate audio feature nodes to obtain the audio feature $\hat{\mathbf{X}}$ with the contextual association

$$\hat{\mathbf{X}} = \hat{\mathbf{A}}\mathbf{X}\mathbf{W}_{\phi}^{\top} + \mathbf{X}. \tag{3}$$

Here, $\hat{\mathbf{X}}$ is the output audio feature of the audio encoder, which contains the timing information of the audio signal because the learnt relations of the aggregated audio feature nodes reflect the contextual association within the audio signal.

The node aggregation with the learnt adjacency graph can highlight the important semantic information about acoustic scenes and events with the time dependency of the audio signal. With the residual connection, the obtained graph audio feature represents long-time dependency information from graph attention and local-dependency information from PANNs with external knowledge via model pretraining.

C. Transformer Decoder

To generate captions, we use the Transformer decoder which takes the audio feature with the contextual association. The decoder has two inputs. One is the audio feature with contextual association $\hat{\mathbf{X}}$, the other is the word embedding with positional encoding $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, \cdots, \mathbf{y}_{N-1}]^{\top} \in \mathbb{R}^{N\times D}$, where N denotes the number of words in the caption. Noting that, $\mathbf{y}_0$ is from the token $<$ sos $>$ representing the start of the sequence, and it does not belong to the caption. The n-th word $\mathbf{y}_n$ in the caption is generated as

$$p(\mathbf{y}_n|\hat{\mathbf{X}}, \mathbf{Y}_{pre}) = \text{Decoder}(\hat{\mathbf{X}}, \mathbf{Y}_{pre}), \tag{4}$$

where $p(\mathbf{y}_n|\hat{\mathbf{X}}, \mathbf{Y}_{pre})$ is the posterior probability of the n-th word $\mathbf{y}_n$ and $1 \le n \le N$. $\mathbf{Y}_{pre} = [\mathbf{y}_0, \cdots, \mathbf{y}_{n-1}]^{\top}$ denotes the previously generated caption words.

We pretrain a Word2Vec language model [20] by combining the captions on the Clotho-v2 dataset [1] and an external dataset, i.e., AudioCaps [21], for an effective word embedding. Moreover, the proposed GraphAC method is firstly pretrained on the AudioCaps dataset and then fine-tuned on the Clotho-v2 dataset, with the same training strategy as in [2], [22].

III. Experiments and Results

A. Dataset

Following Task 6 of the DCASE 2021 Challenge and related work, such as [2], we use the development and validation splits of the AudioCaps dataset [21] for pretraining, the development and validation splits of the Clotho-v2 dataset [1] for fine-tuning, and the evaluation split of the Clotho-v2 dataset for evaluation. Specifically, AudioCaps has 44366, 458, and 905 audio clips in the development, validation, and evaluation sets, and Clotho-v2 has 3839, 1045 and 1045 audio clips in the development, validation and evaluation splits, respectively.

B. Experimental Setup

In the graph attention module, k was empirically set as 25 in the top-k mask. Following P-Transformer [2], GraphAC applied SpecAugment and mix-up strategies to improve generalisation. The cross-entropy loss with label smoothing [23] was used with Adam optimizer [24] to optimize the network. The batch size was 16, and the learning rate was 0.0001. The decoder used a teacher forcing strategy in training and a beam search strategy with a beam size of 5 in evaluation.

Following DCASE Challenge, all the methods are evaluated by machine translation metrics (i.e., BLEU$_n$, ROUGE$_l$ and METEOR) and captioning metrics (CIDE$_r$, SPICE and SPIDE$_r$). BLEU$_n$ [25] measures a modified n-gram precision. ROUGE$_l$ [26] is a score based on the longest common subsequence. METEOR [27] is a harmonic mean of weighted unigram precision and recall. CIDE$_r$ [28] is a weighted cosine similarity of n-grams. SPICE [29] is the F-score of semantic propositions extracted from caption and reference. SPIDE$_r$ [30] is the mean score between CIDE$_r$ and SPICE, which evaluates both the fluency and semantic properties of the caption. The source code[1] along with examples of the predicted captions is released for reproducibility of our work.

C. Performance Comparison

We compare the proposed method with the state-of-the-art methods that all use PANNs as the encoder to extract the audio feature but do not model the long-time dependencies, including P-Transformer [2] (backbone method), SJTU [3], P-Conformer [12], CNN14-M2Transformer [13], MAAC [14] and EaseAC [9]. All these methods adopt Word2Vec in the decoder to obtain the word embedding for caption prediction, except EaseAC and P-Conformer. Since reinforcement learning [31] is not employed in the proposed GraphAC, for fair comparisons, it is not used in any of the compared methods in our experiments. Note that, without reinforcement learning does not affect the main conclusion drawn in the comparisons. For a fair comparison, EaseAC is pretrained on the AudioCaps dataset without using the private dataset in [9].

Table I shows the performances of the proposed GraphAC method and the state-of-the-art methods. The proposed GraphAC outperforms these state-of-the-art methods in all evaluation metrics, including SPIDE$_r$, the most important caption metric in the ranking of the DCASE Challenge. Different from other methods, the proposed GraphAC method models the long time dependencies of the audio feature through graph attention. The result shows the effectiveness of the proposed GraphAC method and the importance of modelling long-range temporal information in audio feature representation for the audio captioning task. Without the graph attention module, GraphAC is reduced to the backbone method (i.e., P-Transformer).

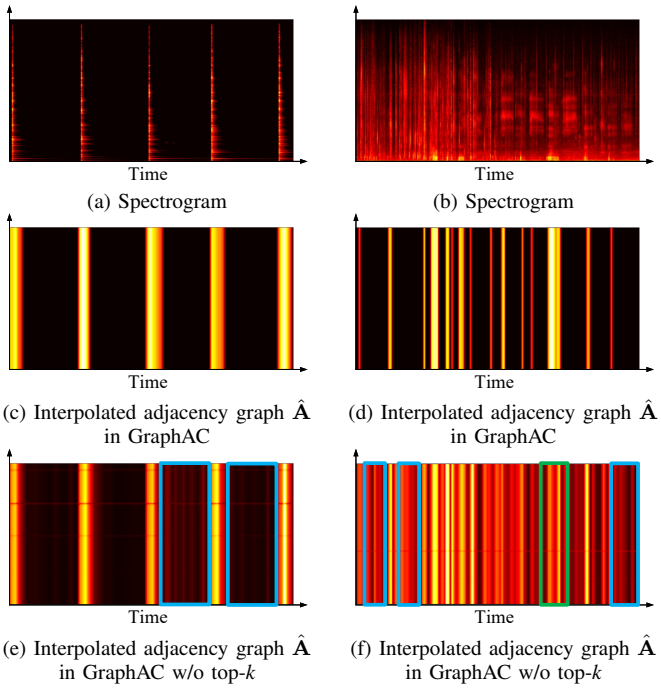
D. Effect of Graph Attention on Audio Feature Representation

Fig. 2 illustrates the mechanism of the graph attention for the audio feature representation, with two audio clips as

[1] https://github.com/LittleFlyingSheep/GraphAC

TABLE I
PERFORMANCE COMPARISON ON THE EVALUATION SPLIT OF THE CLOTHO-V2 DATASET.

Method	$\text{BLEU}_1(\%)$	$\text{BLEU}_2(\%)$	$\text{BLEU}_3(\%)$	$\text{BLEU}_4(\%)$	$\text{ROUGE}_l(\%)$	METEOR(%)	$\text{CIDE}_r(\%)$	SPICE(%)	$\text{SPIDE}_r(\%)$
SJTU [3]	56.5	-	-	15.5	37.4	17.4	39.9	11.9	25.9
P-Conformer [12]	54.1	34.6	23.1	15.2	35.6	16.1	36.2	11.0	23.6
CNN14-M2Transformer [13]	55.5	35.7	23.6	15.3	36.6	16.8	40.9	12.0	26.5
MAAC [14]	57.7	-	-	17.4	37.7	17.4	41.9	11.9	26.9
EaseAC [9]	55.4	35.6	23.5	15.3	36.4	16.7	40.5	11.7	26.1
P-Transformer (backbone) [2]	56.1	37.4	25.7	17.4	37.9	17.1	42.6	12.4	27.5
GraphAC w/o top-k	58.0	**38.8**	**26.5**	17.7	38.4	**17.8**	43.5	12.4	27.9
GraphAC	**58.1**	38.6	**26.5**	**18.1**	**38.5**	17.5	**43.7**	**12.6**	**28.1**



(a) Spectrogram (b) Spectrogram

(c) Interpolated adjacency graph $\hat{\mathbf{A}}$ in GraphAC (d) Interpolated adjacency graph $\hat{\mathbf{A}}$ in GraphAC

(e) Interpolated adjacency graph $\hat{\mathbf{A}}$ in GraphAC w/o top-k (f) Interpolated adjacency graph $\hat{\mathbf{A}}$ in GraphAC w/o top-k

Fig. 2. Audio feature representation. The left column is a discrete sound "five different sounding bells are ringing between short pauses," and the right column is a continuous sound "a small dog snoring and groaning". (a) and (b) are their spectrograms. (c) and (d) are their interpolated adjacency graphs of GraphAC. (e) and (f) are their interpolated adjacency graphs of GraphAC without top-k mask (GraphAC w/o top-k). The blue contours denote the meaningless areas with over attention, and the green contour denotes the important area with insufficient attention by the GraphAC w/o top-k.

examples. The clip in the left column is a transient sound of "five different sounding bells are ringing between short pauses," and the clip in the right column is a continuous sound of "a small dog snoring and groaning". Their spectrograms are shown in Fig. 2(a) and (b). The corresponding adjacency graphs $\hat{\mathbf{A}}$ learnt with the top-k mask are in Fig. 2(c) and (d), and those without the top-k mask are in Fig. 2(e) and (f), with bilinear interpolation applied for better illustration.

It can be found that the interpolated adjacency graph displays the vertical patterns. The most possible reason is that the graph attention mechanism employs an additive score function (i.e., Eq. (1) normalised by softmax function) to obtain the attention coefficients between the audio nodes. It focuses on whether the nodes contain important information about the scenes and events, with greater attention coefficients implying more important nodes than others.

The interpolated adjacency graph highlights audio feature nodes with long-time dependency on acoustic scenes and events. As shown in Fig. 2(c) and (d), it works for both transient and continuous sound. The highlighted audio feature nodes have a vertical bar pattern in the adjacency graph. This is because the graph modelling in GraphAC is applied on $\mathbf{x}_i, 1 \leq i \leq T$, i.e., the audio feature node at each time frame. The learnt node relations are reflected as the asymmetric directed adjacency graph $\hat{\mathbf{A}}$, which highlights time-dependencies between audio nodes. The selected audio feature nodes by the adjacency graph are those with long-time dependencies among all nodes that can help capture the contextual information from the audio signal, and thus the semantic information about acoustic scenes and events.

E. Effect of the Top-k Mask

We compared GraphAC and GraphAC without the top-k mask in graph attention (GraphAC w/o top-k) in Table I. Results show that the performance without the top-k mask degrades in core semantic metrics, i.e., CIDE_r, SPICE and SPIDE_r. Examples of their adjacency graphs (bilinear interpolated) are shown in Fig. 2(c)-(f). The adjacency graph generated by GraphAC w/o top-k has attention to the meaningless background audio feature nodes, contoured in blue in Fig. 2(e) and (f), and insufficient attention to some important nodes, contoured in green in Fig. 2(f). In contrast, the proposed method with the top-k mask can focus on the important audio feature nodes and ignore the meaningless audio feature nodes, when modelling the audio feature with long-time dependencies. In this work, $k = 25$ is selected empirically from the experiments. Future work will include the adaptive estimation of the k value from audio signals with different time duration.

IV. CONCLUSION

We have presented a novel audio captioning method using graph attention in the encoder to exploit temporal dependencies in audio features. It facilitates the decoder in generating better captions with the timing and contextual information of the audio signal. Experiments show that the proposed method achieves state-of-the-art captioning performance. In addition, the proposed graph modelling enables audio feature representation with temporal information, which may benefit other tasks such as audio scene classification and event detection. Future work will investigate the latent relationship between audio feature nodes and the caption words by the graph learning, and develop methods for estimating k in the top-k mask adaptively to suit audio objects with different time spans.

REFERENCES

[1] K. Drossos, S. Lipping, T. Virtanen, "Clotho: An audio captioning dataset", in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, Barcelona, Spain, 2020, pp. 736–740.

[2] X. Mei, Q. Huang, X. Liu, G. Chen, J. Wu, Y. Wu, J. Zhao, S. Li, T. Ko, H. L. Tang, X. Shao, M. D. Plumbley, W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning", in *Proc. DCASE Workshop*, 2021.

[3] X. Xu, Z. Xie, M. Wu, K. Yu, "The SJTU system for DCASE2021 Challenge Task 6: Audio captioning based on encoder pre-training and reinforcement learning", DCASE2021 Challenge, Tech. Rep., Jul. 2021.

[4] K. Drossos, S. Adavanne, T. Virtanen, "Automated audio captioning with recurrent neural networks", in *Proc. IEEE Workshop Appl. Signal Process Audio Acoust.*, New Paltz, NY, USA, 2017, pp. 374–378.

[5] E. Cakır, K. Drossos, T. Virtanen, "Multi-task regularisation based on infrequent classes for audio captioning" in *Proc. DCASE Workshop*, 2020.

[6] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, S. Saito, "A Transformer-based audio captioning model with keyword estimation", in *Proc. INTERSPEECH*, 2020.

[7] M. Schuster, K. K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Trans. on Signal Process.*, vol. 45, no. 11, pp.2673-2681. Nov., 1997.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, "Attention is all you need", in *Proc. Advances in Neural Information Processing Systems*, 2017.

[9] Q. Han, W. Yuan, D. Liu, X. Li, Z. Yang, "Automated audio captioning with weakly supervised pre-training and word selection methods", in *Proc. DCASE Workshop*, 2021.

[10] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, M. D. Plumbley, "PANNs: Large-scale pretrained audio neural networks for audio pattern recognition", *IEEE/ACM Trans. Audio Speech Lang.*, vol. 28, pp. 2880–2894, Nov., 2020.

[11] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, M. Ritter, "AudioSet: An ontology and human-labeled dataset for audio events", in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, New Orleans, LA, USA, 2017, pp. 776–780.

[12] C. Narisetty, T. Hayashi, R. Ishizaki, S. Watanabe, K. Takeda, "Leveraging state-of-the-art ASR techniques to audio captioning", DCASE2021 Challenge, Tech. Rep., Jul. 2021.

[13] Z. Chen, D. Zhang, J. Wang, F. Deng, "Audio captioning with meshed-memory Transformer", DCASE2021 Challenge, Tech. Rep., Jul. 2021.

[14] Z. Ye, H. Wang, D. Yang, Y. Zou, "Improving the performance of automated audio captioning via integrating the acoustic and semantic information", in *Proc. DCASE Workshop*, 2021.

[15] X. Mei, X. Liu, M. D. Plumbley, W. Wang, "Automated audio captioning: an overview of recent progress and new challenges", arXiv preprint arXiv:2205.05949, 2022.

[16] H. Song, S. Deng, J. Han, "Exploring inter-node relations in CNNs for environmental sound classification", *IEEE Signal Process. Lett.*, vol. 29, pp. 154–158, Nov., 2021.

[17] F. Huang, Z. Li. "Improve image captioning via relation modeling", in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, Singapore, 2022, pp. 1945–1949.

[18] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, Z. Zha. "Object relational graph with teacher-recommended learning for video captioning", in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, USA, 2020, pp. 13278–13288.

[19] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, "Graph attention networks", in *Proc. Int. Conf. Learn. Represent.*, 2018.

[20] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient estimation of word representations in vector space", in *Proc. Int. Conf. Learn. Represent.*, Scottsdale, AZ, USA, 2013.

[21] C. D. Kim, B. Kim, H. Lee, G. Kim, "Audiocaps: Generating captions for audios in the wild", in *Proc. Conf. N. Am. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Minneapolis, MN, USA, 2019, pp. 119–132.

[22] F. Xiao, J. Guan, H. Lan, Q. Zhu, W. Wang, "Local information assisted attention-free decoder for audio captioning", *IEEE Signal Process. Lett.*, vol. 29, pp. 1604–1608, Jul., 2022.

[23] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, "Rethinking the inception architecture for computer vision", in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2818–2826.

[24] D. P. Kingma, J. Ba, "Adam: A method for stochastic optimization", in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, 2015.

[25] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation", in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, 2002, pp. 311–318.

[26] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries", in *Proc. Text Summarization Branches Out*, 2004, pp. 74–81.

[27] A. Lavie, A. Agarwal, "METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments", in *Proc. Annu. Meet. Assoc. Comput. Linguist.*, Prague, Czech republic, 2007, pp. 228–231.

[28] R. Vedantam, C. Lawrence Zitnick, D. Parikh, "CIDEr: Consensus-based image description evaluation", in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Boston, MA, USA, 2015, pp. 4566–4575.

[29] P. Anderson, B. Fernando, M. Johnson, S. Gould, "SPICE: Semantic propositional image caption evaluation", in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 382–398.

[30] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, K. Murphy, "Improved image captioning via policy gradient optimization of SPIDEr", in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 873–881.

[31] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, V. Goel, "Self-critical sequence training for image captioning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Honolulu, Hawaii, USA, 2017, pp. 7008–7024.