IMPROVING THE PERFORMANCE OF AUTOMATED AUDIO CAPTIONING VIA INTEGRATING THE ACOUSTIC AND SEMANTIC INFORMATION

Zhongjie ye[1]*, Helin Wang*[1]*, Dongchao Yang*[1]*, Yuexian Zou*[1,2,*]

[1] ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2] Peng Cheng Laboratory, Shenzhen, China
{zhongjieye@stu.pku.edu.cn, wanghl15@pku.edu.cn
dongchao98@stu.pku.edu.cn, zouyx@pku.edu.cn}

ABSTRACT

Automated audio captioning (AAC) has developed rapidly in recent years, involving acoustic signal processing and natural language processing to generate human-readable sentences for audio clips. The current models are generally based on the neural encoder-decoder architecture, and their decoder mainly uses acoustic information that is extracted from the CNN-based encoder. However, they have ignored semantic information that could help the AAC model to generate meaningful descriptions. This paper proposes a novel approach for automated audio captioning based on incorporating semantic and acoustic information. Specifically, our audio captioning model consists of two sub-modules. (1) The pre-trained keyword encoder utilizes pre-trained ResNet38 to initialize its parameters, and then it is trained by extracted keywords as labels. (2) The multi-modal attention decoder adopts an LSTM-based decoder that contains semantic and acoustic attention modules. Experiments demonstrate that our proposed model achieves state-of-the-art performance on the Clotho dataset. Our code can be found at `https://github.com/WangHelin1997/DCASE2021_Task6_PKU`.

Index Terms— Audio captioning, pre-training, multi-modal attention, keyword classification

1. INTRODUCTION

Automated audio captioning (AAC) is a cross-modal task of generating a natural language description for an audio clip. It is different from audio tagging (AT), acoustic scene classification (ASC) and automatic speech recognition. The purpose of AAC is not only to analyze acoustic scenes, events, and concepts in a given audio clip, but also to find the relationships among them to produce human-readable sentences. Applications of automated audio captioning are diverse such as assisting the hearing impaired people by converting audio signals into a text, and content-based audio retrieval task which uses the free-form natural language queries to retrieve the audio [1].

AAC has aroused a lot of interest among researchers since the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 challenge. Nowadays, the mainstream framework is based on neural encoder-decoder systems which have achieved success in some relevant fields such as image captioning [2]. The current AAC models consist of a convolutional neural network (CNN) encoder and a recurrent neural network (RNN) (or Transformer) decoder with an attention mechanism. The inputs used could be log-mel energies, Mel-Frequency Cepstral Coefficients(MFCCs), or other acoustic features which are extracted from raw audio clips. They are firstly encoded by a CNN encoder into a set of feature vectors. Then, they are decoded into sentences by an RNN-based or Transformer-based decoder with (or without) an attention mechanism.

Over past few years, there are amounts of methods proposed in AAC task [3, 4, 5, 6, 7] based on neural encoder-decoder systems. M. Wu *et al.* [3] straightly takes the mean of the feature vectors that are the outputs of the encoder in the time dimension, and uses them as the input of the decoder. H. Wang *et al.* [5] proposed a temporal attention mechanism in the decoder, which could utilize more acoustic information for each time step. In contrast to previous work in AAC, Y. Wu *et al.* [4] and X. Xu *et al.* [6] explore transfer learning method to help AAC models to get better performance. The strategy of their proposed methods could be divided into two stages. In the first stage, a tagging system is pre-trained by ASC or AT task. Then the parameters of the audio encoder are initialized by the pre-trained tagging system. In the second stage, the whole AAC model is trained end-to-end by minimizing the cross-entropy (CE) loss. With these methods mentioned above, they generally only consider acoustic information while ignoring semantic information when the AAC model generates sentences. Specifically, the semantic information could contain keywords that are from the encoder, previously predicted words in the decoding time, and so on. In this paper, we introduce semantic information with acoustic information to assist the decoder to generate higher quality sentences. Furthermore, to better make use of semantic and acoustic information, we propose a novel multi-modal attention mechanism. In summary, our contributions are as follows:

1. We propose a **m**ulti-modal **a**ttention-based **a**udio **c**aptioning model with a pre-trained keyword encoder, named **MAAC**. It could utilize both acoustic and semantic information to generate the description. The semantic information includes keywords from the pre-trained keyword encoder and the previously decoding information from the decoder.
2. Our MAAC achieves a new state-of-the-art performance on the Clotho dataset. We present the ablation analysis of the components of our MAAC and demonstrate that semantic information could improve the performance of the AAC model.

The organization of the paper is as follows. Section 2 introduces our proposed model. We present our experimental results

*Yuexian Zou is the corresponding author.

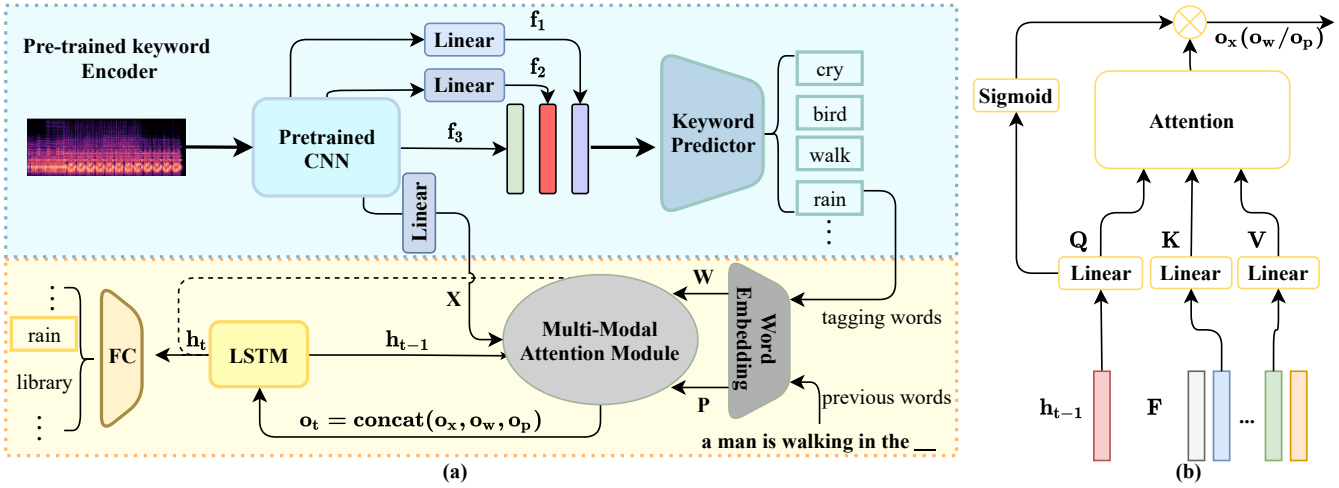

Figure 1: (a) Our proposed MAAC includes two submodules: the pre-trained keyword encoder is on the top and the LSTM-based decoder with a multi-modal attention module is on the bottom. (b) The architecture of the attention mechanism. F could represent acoustic features or semantic features.

and evaluations in Section 3. Finally, we give concluding remarks and possible future directions in Section 4.

2. SYSTEM ARCHITECTURE

In this section, our proposed MAAC is introduced and its architecture is shown in Figure 1. Specifically, our MAAC consists of two submodules: a pre-trained keyword encoder and an LSTM-based decoder with a multi-modal attention module. In the following subsections, we will introduce details about it.

2.1. Pre-trained Keyword Encoder

The CNN encoder, which is widely used in the AAC challenge [4, 5], plays an important role in extracting acoustic information from raw audios. In this work, we extract keywords from captions as training labels and use the pre-trained ResNet38[1] [8] that performs well in the AudioSet dataset [9] as our backbone network.

Constructing Audio-Keyword Training Pairs Firstly, Natural Language Toolkit (NLTK[2]) is a powerful open-source tool applied to extract words from each caption. We choose the nouns and verbs to construct the keyword table by getting rid of some useless words such as *make*, *go*, *others*, etc. The verbs in the keyword table are transformed into their original forms and the nouns are not changed, because plural forms of the nouns have different meanings. Then, we choose N keywords with the highest frequency from the modified keyword table and use them as labels for pre-training.

We combine all the keywords from the 5 captions of each audio clip to form the training label which is a multi-hot vector. Each word of captions is transformed into its original forms according to the above rules. When a word occurs in the keyword table, the corresponding position of the multi-hot vector is set to 1, otherwise 0.

[1] https://github.com/qiuqiangkong/audioset_tagging_cnn

[2] https://github.com/nltk/nltk

Training the Keyword Encoder As Figure 1 illustrates, the pre-trained ResNet38 is used as our backbone, which consists of 6 convolutional blocks. We refine it with a feature hierarchy structure to combine multi-level features, *i.e.* the features after the third, fourth, and last convolution block. Then all of them are passed into different linear layers after the global average pooling (GAP) method to obtain f_1, f_2 and f_3. Finally, we use them to obtain the predictions $\hat{y} \in \mathbb{R}^N$ and N is the number of keywords.

$$\hat{y} = \sigma(Linear(concat(f_1, f_2, f_3))) \tag{1}$$

where σ denotes sigmoid activation function. Given the ground-truth $y \in \mathbb{R}^N$, the pre-trained keyword encoder could be optimized by minimizing the binary cross-entropy loss:

$$\mathcal{L}_{bce}(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y(i) log\ \hat{y}(i) \tag{2}$$

2.2. Multi-modal Attention Decoder

Unlike the existing audio captioning models, we further incorporate acoustic with semantic information into generating captions: we propose a multi-modal attention module to incorporate them. The high-level representation of acoustic information denoted as $\boldsymbol{X} = \{x_1, ..., x_L\} \in \mathbb{R}^{L \times C_1}$, is the output of a linear layer whose input is the output of the last convolution block of the pre-trained keyword encoder. The semantic features contain the keywords $\boldsymbol{W} = \{w_1, ..., w_K\}$ that is the K outputs of the pre-trained keyword encoder, and the previously predicted words $\boldsymbol{P} = \{p_1, ..., p_{t-1}\}$ that contain all the generated words before time step t. Both of them are transformed into continuous vectors by a randomly initialized embedding layer **Emb**, $\boldsymbol{W} \in \mathbb{R}^{K \times C_2}$ and $\boldsymbol{P} \in \mathbb{R}^{(t-1) \times C_2}$. The implementation process of the multi-modal attention module is as follows.

Firstly, all of them are transformed into the same latent space, where X is turned to $\hat{\boldsymbol{X}} \in \mathbb{R}^{T \times C}$, W becomes $\hat{\boldsymbol{W}} \in \mathbb{R}^{K \times C}$ and P becomes $\hat{\boldsymbol{P}} \in \mathbb{R}^{(t-1) \times C}$. Then the hidden states as intermediaries

Table 1: Single-model performances on the Clotho [10] evaluation splits in the CE and RL training period. B1, B4, RG, ME, CD, SP, and SD denote BLEU-1, BLEU-4, METEOR, ROUGE-L, CIDEr-D, SPICE, and SPIDEr, respectively. For all metrics, higher values indicate better performance.

	Cross-entropy							CIDEr-D optimization						
Model	B1	B4	RG	ME	CD	SP	SD	B1	B4	RG	ME	CD	SP	SD
Baseline [10]	37.8	1.7	26.3	7.8	7.5	2.8	5.1	-	-	-	-	-	-	-
TAM [5]	48.9	10.7	32.5	14.8	25.2	9.1	17.2	-	-	-	-	-	-	-
TM [4]	53.4	15.1	35.6	16.0	34.6	10.8	22.7	-	-	-	-	-	-	-
UNIS's model [11]	-	-	-	-	-	-	-	62.5	17.8	40.1	17.6	42.8	12.6	27.7
SJTU's model [12]	56.5	15.5	37.4	17.4	39.9	11.9	25.9	64.0	16.3	40.4	17.8	44.9	12.3	28.6
MAAC (Ours)	57.7	17.4	37.7	17.4	41.9	11.9	26.9	**64.8**	**18.1**	**40.8**	**19.0**	**49.1**	**13.1**	**31.1**

Table 2: Settings and results of ablation studies. The results are reported after CE training stage. SAM denotes the semantic attention module.

Model	B4	CD	SD
Base	16.5	40.6	26.4
+ Previously predicted words	17.1	41.1	26.4
+ Keywords	can not converge		
+ Both (w/o sharing SAM)	16.8	41.1	26.7
proposed MAAC	**17.4**	**41.9**	**26.9**

connect $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{P}}$, through a multi-modal attention mechanism that is shown in Figure 2. Taking the acoustic information for example: given the previous LSTM hidden state h_{t-1}, we use a single fully-connected layer followed by a softmax function to generate the attention distributions α of acoustic features in the time dimension. Finally, the gated linear unit (GLU) is applied to the output of the attention module, to control how much information should flow into the next layer. Formula (3)-(5) are the definitions of the acoustic attention module $\boldsymbol{\Psi_x}$:

$$\boldsymbol{A} = ReLU((\hat{\boldsymbol{X}}\boldsymbol{W_i^T} + b_i) \oplus (h_{t-1}\boldsymbol{W_s^T} + b_s)) \tag{3}$$

$$\alpha = softmax(\boldsymbol{A}\boldsymbol{W_n} + b_n) \tag{4}$$

$$o_x = GLU([\hat{\boldsymbol{X}} \otimes \boldsymbol{\alpha}, h_{t-1}]) \tag{5}$$

where $\boldsymbol{W_s} \in \mathbb{R}^{M\times H}$, $\boldsymbol{W_i} \in \mathbb{R}^{M\times C}$, $\boldsymbol{W_n} \in \mathbb{R}^{M}$ are transformation matrixes that map acoustic features and hidden states to the same dimension. Here are $b_s \in \mathbb{R}^M$, $b_i \in \mathbb{R}^M$, and $b_n \in \mathbb{R}^1$. We denote $\oplus$ as the element-wise addition of a matrix and a vector, and $\otimes$ as the element-wise multiplication of a matrix and a vector. We choose the GLU operation to obtain the output $o_x \in \mathbb{R}^C$, which implements a simple gating mechanism over the output $\mathcal{Y} = [\mathcal{A}, \mathcal{B}] \in \mathbb{R}^{2d}$:

$$GLU([\mathcal{A}, \mathcal{B}]) = \mathcal{A} \otimes \sigma(\mathcal{B}) \tag{6}$$

where $\mathcal{A} \in \mathbb{R}^d, \mathcal{B} \in \mathbb{R}^d$ are the inputs to the non-linearity, and the output $GLU([\mathcal{A}, \mathcal{B}]) \in \mathbb{R}^d$ is half the size of $\mathcal{Y}$ [13].

As for the semantic information, the same structure of the attention module is applied to keywords and previously predicted words, and the outputs are $o_w \in \mathbb{R}^C$ and $o_p \in \mathbb{R}^C$ respectively. Note each part of semantic information shares an attention module. We add o_x, o_w, o_p with w_{t-1} which is a predicted word of the last time step to obtain the output o_t. Then, o_t and h_{t-1} are sent to calculate the hidden state h_t which is used to predict word probability distribution v_t. Finally, the current word w_t is chosen from v_t with the highest probability and added to previously predicted words $\boldsymbol{P}$ for the next iteration of LSTM. Formula (7) is the operation of the multi-modal attention module described above:

$$\begin{aligned} h_0 &= GAP(\hat{\boldsymbol{X}}) \\ h_t &= LSTM(h_{t-1}, Add(o_x, o_w, o_p, \mathbf{Emb}(w_{t-1}))) \\ v_t &= Softmax(Linear(h_t)) \end{aligned} \tag{7}$$

where h_0 represents the global information of acoustic features in the time dimension. $v_t \in \mathbb{R}^{|\Sigma|}$ is a probability vector, and $|\Sigma|$ is a predefined dictionary including all words.

3. EXPERIMENT

3.1. Dataset and Experiment Setup

Clotho v2 We evaluate our proposed method on the Clotho v2 dataset [10], which is published in DCASE 2020 and expanded in DCASE 2021. Nowadays it contains 5,929 audio clips labeled with 5 captions for each, including 3,839 training, 1,045 validation, and 1,045 testing audio clips. We convert all sentences to lower case and remove all punctuation marks, ending up with a vocabulary $|\Sigma|$ of 4368 words including special tokens "BOS", "EOS", and "PAD". For evaluation, we employ standard evaluation metrics: BLEU [14], ROUGE-L [15], METEOR [16], CIDEr-D [17], SPICE [18] and SPIDEr that is the mean of CIDEr-D and SPICE. All metrics are computed with the audio captioning evaluation tool[3].

Implementation Details We choose $N = 300$ keywords for pre-training encoder, $K = 5$ keywords and the dimension of fully-connected layers C_1, C_2 and C are 512. The decoder LSTM has 512 hidden units, word embedding size is also set to 512. To mitigate overfitting, dropout regularization [19] is used in the word embedding layer with a rate of 0.5, and the word classification layer with a rate of 0.25.

The training strategy of the MAAC could be divided into two stages: encoder pre-training and the whole MAAC model training. In the phase of training the encoder, firstly the CNN backbone is frozen up, trained with the initial learning rate of 1×10^{-3} for 80

[3] https://github.com/audio-captioning/caption-evaluation-tools

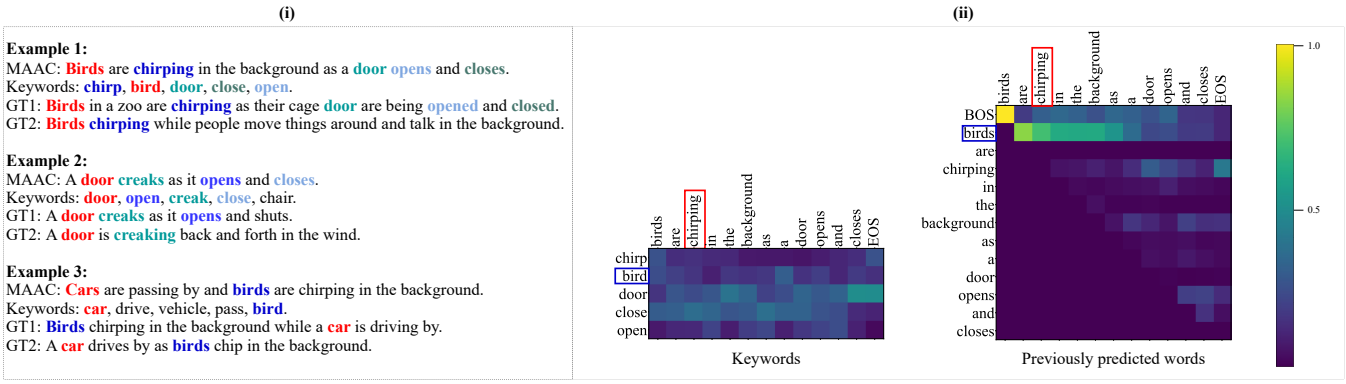

Figure 2: (i) It shows some examples of MAAC outputs and colored words indicate that keywords appear in both predicted and ground-truth sentences. (ii) The visualization for attention matrices of keywords and previously predicted words in the semantic attention module of example 1.

epochs. Next, we finetune the whole keyword encoder with the learning rate of 5×10^{-4} for 25 epochs. Then, it can be divided into two parts for training the whole MAAC: CE training and RL fine-tuning. CE training takes 30 epochs while the parameters of the pre-trained keyword encoder are frozen. Finally, the 30^{th} CE training model is used for reinforcement learning (RL) fine-tuning 55 epochs. In all training stages, we adopt an Adam optimizer with a mini-batch size of 32, and exponential decay to adjust the learning rate with a factor of 0.98 every epoch. The initial learning rates are set to 3×10^{-4} and 5×10^{-5} for two parts of training the whole MAAC. In the inference stage, we adopt beam search with a beam size of 4 that is implemented to achieve the best decoding performance.

In order to avoid over-fitting and increase data diversity, SpecAugment [20], SpecAugment++ [21], Mixup [22], Label smoothing [23] and teacher forcing [24] are used in the training phase. For Mixup method, it is just used in the training of the keyword encoder. The label smoothing and teacher forcing are just used while training the whole MAAC.

3.2. Result Analysis

We compare our proposed MAAC with the following current models: (1) Baseline [10] is proposed by K. Drossos *et al.*, which employs a GRU-GRU encoder-decoder framework; (2) Temporal attention model (TAM) [5] uses the CNN encoder and the LSTM-based decoder with the temporal attention mechanism; (3) Transformer-based model (TM) [4] adopts a pre-training strategy to improve captioning performance; (4) UNIS's model [11] uses PANNs to initialize the parameters of the encoder and is pre-trained on AudioCaps dataset [25]; (5) SJTU's model [12] utilizes AudioSet to pre-train its encoder in order to enhance the ability of the encoder to recognize audio concepts. Both (5) and (6) adopt RL training to obtain the final models.

Table 1 lists the results of various single models on the Clotho dataset. Our MAAC achieves the highest score on all metrics in the CIDEr-D optimization stage. In addition, the CIDEr-D score of the proposed MAAC improves from 41.9 to 49.1 after further optimizing CIDEr-D.

Through Figure 2 (i), we can find that the pre-trained keyword encoder can almost recognize the main concepts *i.e.* keywords (e.g. *bird* and *chirp* in example 1) of a given audio clip, and the keywords may appear in different states in the ground-truth captions and the predicted sentences. Figure 2 (ii) further shows that keywords and previously predicted words are concerned to generate the current word. For instance, when the decoder is generating "chirping", it pays more attention to the "birds" in the previously predicted words but pays less attention to "birds" in the keywords. That is to say, previously predicted words and keywords are complementary to each other in the semantic attention module.

3.3. Ablative Analysis

To quantify the impact of the proposed multi-modal attention module, we compare our MAAC against a set of other ablated models with different settings. The results of various models are shown in Table 2. We firstly design the "base" model which does not use previously predicted words and keywords (*i.e.* the semantic attention module). Then we add the information of previously predicted words or keywords to the "base" model. We find that it has little impact on the performance of the model by only introducing previously predicted words. It might be that previously predicted words would contain wrong words that destroys the input information of the decoder. In addition, the model which only uses the keywords in the semantic attention module could not converge. From section 3.2, we know that keywords contain the main concepts of an audio clip. When we only utilize them in the semantic attention module, they will cause the decoder to pay more attention to the part of the keywords and ignore the overall semantic relationship. Moreover, we examine the performance of using a shared (or not) semantic attention module on its performance and find that a sharing semantic attention module could further improve the CIDEr-D score.

4. CONCLUSION

In this paper, we propose a novel audio captioning model based on the multi-modal attention module which utilizes both acoustic and semantic information to generate captions. In addition, the performance of the MAAC achieves a new state-of-the-art under the two stages of training. The ablation experiments further demonstrate the effectiveness of the multi-modal attention module. In future work, we would concentrate on how to align the multi-modal information more effectively to improve the performance of the AAC.

5. REFERENCES

[1] A.-M. Oncescu, A. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," *arXiv preprint arXiv:2105.02192*, 2021.

[2] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*. PMLR, 2015, pp. 2048–2057.

[3] 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.

[4] Y. Wu, K. Chen, Z. Wang, X. Zhang, F. Nian, S. Li, and X. Shao, "Audio captioning based on transformer and pre-training for 2020 DCASE audio captioning challenge," DCASE2020 Challenge, Tech. Rep., 2020.

[5] H. Wang, B. Yang, Y. Zou, and D. Chong, "Automated audio captioning with temporal attention," DCASE2020 Challenge, Tech. Rep., 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] 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.

[8] 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.

[9] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[10] 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.

[11] 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, and W. Wang, "An encoder-decoder based audio captioning system with transfer and reinforcement learning for DCASE challenge 2021 task 6," DCASE2021 Challenge, Tech. Rep., July 2021.

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

[13] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1243–1252.

[14] 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.

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

[16] 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.

[17] 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.

[18] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *European Conference on Computer Vision*. Springer, 2016, pp. 382–398.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] 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.

[21] H. Wang, Y. Zou, and W. Wang, "Specaugment++: A hidden space data augmentation method for acoustic scene classification," *arXiv preprint arXiv:2103.16858*, 2021.

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

[23] 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.

[24] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015.

[25] 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.