A CRNN-GRU BASED REINFORCEMENT LEARNING APPROACH TO AUDIO CAPTIONING

Xuenan Xu, Heinrich Dinkel, Mengyue Wu, Kai Yu

MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
AI Institute, Shanghai Jiao Tong University, Shanghai, China
{*wsntxxn, richman, mengyuewu, kai.yu*}@sjtu.edu.cn

ABSTRACT

Audio captioning aims at generating a natural sentence to describe the content in an audio clip. This paper proposes the use of a powerful CRNN encoder combined with a GRU decoder to tackle this multi-modal task. In addition to standard cross-entropy, reinforcement learning is also investigated for generating richer and more accurate captions. Our approach significantly improves against the baseline model on all shown metrics achieving a relative improvement of at least 34%. Results indicate that our proposed CRNN-GRU model with reinforcement learning achieves a SPIDEr of 0.190 on the Clotho evaluation set[1]. With data augmentation, the performance is further boosted to 0.223. In the DCASE challenge Task 6 we ranked fourth based on SPIDEr, second on 5 metrics including BLEU, ROUGE-L and METEOR, without ensemble or data augmentation while maintaining a small model size (only 5 Million parameters).

***Index Terms*—** audio captioning, reinforcement learning, convolutional recurrent neural networks

1. INTRODUCTION

Automatic captioning is a challenging task that involves joint learning of different modalities. For example, image captioning requires extracting features from an image and combining them with a language model to generate reasonable sentences to describe the image. Similarly, video captioning learns features from a temporal sequence of images as well as audio to generate captions. However, audio captioning does not attract much attention [1], unlike in the image and video fields. By its nature, captioning is a novel multi-modal task that captures the fine details within an auditory scene with natural language (text). Unlike other tasks such as sound or acoustic event detection, which only focuses on narrow single-label estimation of an event, audio captioning is concerned with producing rich sentences appropriately and precisely describing an audio. Audio captioning has great potential in real-world applications, such as audio surveillance, automatic content description and content-oriented machine-to-machine interaction.

Initial work in audio captioning has been done in [1], which utilized the commercial ProSound Effects [2] audio corpus as a proof of concept. The paper utilized an encoder-decoder architecture containing a three-layer bidirectional gated recurrent unit (BiGRU) encoder and a two-layer BiGRU decoder. An attention pooling is added to summarize the encoder sentence. Subsequent work in [3] investigated audio captioning within the scope of Chinese captioning, firstly proposing a public captioning corpus, focusing on dialogues within a hospital setting. Their results showed that within a limited domain, audio captions can indeed be generated by a single layer encoder-decoder GRU network successfully, but also questioned if commonly utilized metrics for machine translation can well evaluate the final performance. The main discussion is that even though their approach achieves measurably near-human performance via objective metrics, the generated sentences are often less useful according to human evaluation.

Similar to other text generation tasks like machine translation and image captioning, *exposure bias* also exists in audio captioning. Neural network-based models are typically trained in “teacher forcing” fashion, meaning they aim to maximize the likelihood of a future ground-truth word given the current ground-truth word. However, ground-truth annotations are only available during training, while during inference, the model can solely rely on its own predicted current word to infer the next word. This leads to an error accumulation during test-time. Another problem in text generation tasks is a mismatch between the training objective and evaluation metric. Generative models are typically evaluated by discrete metrics such as BLEU [4], ROUGE-L [5], CIDEr [6] or METEOR [7]. However, these non-differentiable metrics cannot be directly optimized using the standard back-propagation approach.

Previous studies have shown that the application of Reinforcement Learning (RL) can partially circumvent exposure bias while optimizing the discrete evaluation metrics at the same time. RL is first proposed to train natural language generation models in [8]. It takes a generative model as an agent and treats words and context as an external environment. The model parameters define a policy, and the choice of the current generated word corresponds to its action. The reward comes from evaluation scores (BLEU, METEOR, CIDEr etc.) of the sampled sentence. Policy-gradient [9] is used to estimate the gradient of the agent parameters using the reward. Work in [10] improves this method by using rewards from greedy-sampled sentences as the baseline to reduce the high variance of rewards. Subsequent work in [11] also adopts actor-critic methods [12] to estimate the value of generated words instead of sampling from the action space. In this paper, we explore the use of the self-critical sequence training (SCST) approach (proposed in [10]) for audio captioning.

This paper is structured as follows, in Section 2 we put forth our CRNN-based encoder-decoder approach to audio captioning. Then in Section 3, the experimental setup, including front-end features and model parameters, are shown. Our results and analysis are dis-

[1]The code and trained models are available at `https://github.com/wsntxxn/DCASE2020T6`

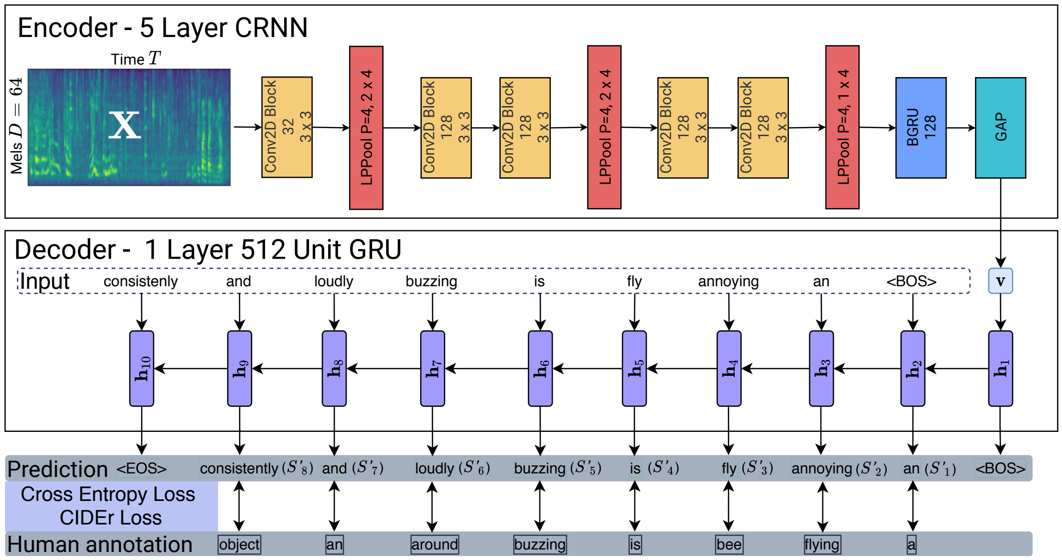

Figure 1: Our proposed encoder-decoder architecture. The encoder is a CRNN model which outputs a fixed-sized 256 dimensional embedding $\mathbf{v}$ after a global average pooling layer (GAP). A convolution block refers to an initial batch normalization, then a convolution, and lastly, a LeakyReLU (slope -0.1) activation. The numbers in each block represent the output channel size and the kernel size. For example, "32, 3×3" means the convolution layer has 32 output channels with a kernel size of 3×3. All convolutions use padding in order to preserve the input size. Then a GRU decoder utilizes this audio embedding $\mathbf{v}$ or embedding of the word $S_t^{'}$ at each time-step, to predict the next word $S_{t+1}^{'}$.

played in Section 4. Lastly, in Section 5 we provide summarization and potential insights from the current work.

2. APPROACH

Similar to previous audio captioning frameworks [3], our approach follows a standard encoder-decoder model (see Equation (1)).

$$\begin{aligned} \mathbf{v} &= \text{Enc}(\mathbf{X}) \\ [S'_1, \dots, S'_T] &= \text{Dec}(\mathbf{v}) \end{aligned} \tag{1}$$

The encoder (Enc) is fed an audio-spectrogram ($\mathbf{X}$) and produces a fixed-sized vector representation $\mathbf{v}$, which the decoder (Dec) uses to predict the caption sentence. Specifically, the decoder generates a single word-token S'_t for each time-step t up until an end of sentence (<EOS>) token is seen (see Figure 1).

In audio captioning, decoding differs between training and evaluation stages:

$$\ell_{\text{XE}}(\theta; S, \mathbf{v}) = -\sum_{t=1}^{T} \log p(S_t|\theta; \mathbf{v}) \tag{2}$$

During training, where transcriptions are available, Dec generates word-tokens given the embedding $\mathbf{v}$ and human-annotated data S, supervised by a cross-entropy (XE) loss (see Equation (2)). During evaluation and testing, no transcriptions are available; thus word-tokens are sampled from the decoder given the audio embedding $\mathbf{v}$. From this description, it is evident that the quality of $\mathbf{v}$ directly affects the generated sentence quality. Thus, our approach mainly diverges from previous approaches in two ways: the encoder architecture and the loss function.

Previous encoder models (GRU) might be insufficient to produce a robust vector representation, thus we replace the standard GRU encoder with a robust convolutional recurrent neural network (CRNN). Our framework can be seen in Figure 1.

Moreover, standard XE training has its potential downsides. For one, the criterion only compares single word-tokens and neglects context information. Second, since each word is treated individually, syntactically incorrect sentences can be generated. Third, optimizing XE inevitably leads to monotonous sentences, because the model is required to precisely imitate a sentence word by word, instead of allowing semantically similar, but different worded sentences.

We employ reinforcement learning for audio captioning. Reinforcement learning allows us to directly back-propagate a metric (e.g., BLEU or CIDEr) in the form of a reward. Formally we train the model to minimize the negative reward of a single sampled sentence S':

$$\ell_{\text{RL}}(\theta; \mathbf{v}) = -r(S'), S' \sim p(S'|\theta; \mathbf{v}) \tag{3}$$

where $S' = [S'_1, S'_2, \dots, S'_T]$. By incorporating the policy gradient method with baseline normalization, the parameter gradients can be estimated as follows:

$$\nabla_\theta \ell(\theta; \mathbf{v}) = -(r(S') - b)\nabla_\theta \log p(S'|\theta; \mathbf{v}), S' \sim p(S'|\theta; \mathbf{v}) \tag{4}$$

here b is a pre-defined baseline normalization constant to reduce the high variance brought by sampling [12]. We set b as the greedy decoding reward because of its effectiveness in image captioning [10].

2.1. Models

Encoder Our proposed encoder is a CRNN model, which has seen success in localizing sound events [13, 14]. The architecture consists of a five-layer CNN (utilizing 3×3 convolutions), summarized into three blocks, with L4-Norm pooling after each block. The CNN blocks subsample the temporal dimension by a factor of 4. A BiGRU is attached after the last CNN output, enhancing our model's ability to localize sounds accurately. At last, we use a global average pooling (GAP) layer in order to remove any time-variability to a single, time-independent representation $\mathbf{v} \in \mathbb{R}^{256}$. The encoder has 679k parameters, making it comparably light-weight while only using 2.7 MB on disk.

Decoder In the context of audio captioning, a decoder takes a fixed-sized embedding and aims to produce a sentence. We use a single-layer GRU with 512 hidden units as our decoder model.

3. EXPERIMENTS

3.1. Dataset

The challenge provides Clotho [2, 15] for the audio captioning task. It contains a total of 4981 audio samples, where the duration is uniformly distributed between 15 to 30 seconds. All audio samples are collected from the Freesound platform. Five native English speakers annotate each sample; thus, 24905 captions are available in total. Captions are post-processed to ensure each caption has eight to 20 words, and the caption does not contain unique words, named entities or speech transcription. The dataset is officially split into three sets, termed as development, evaluation, and testing, with a ratio of 60%-20%-20%. In the challenge, the development and evaluation sets are used for training our audio captioning model while the testing set is for evaluating the model.

3.2. Data pre-processing

We extract 64-dimensional log-Mel spectrogram (LMS) as our default input feature. Here a single frame is extracted via a 2048 point Fourier transform every 20 ms with a Hann window size of 40 ms. This results in a $\mathbf{X} \in \mathbb{R}^{T \times D}$ log-mel spectrogram feature for each input audio, where D = 64 and T is the number of frames. Moreover, the input feature is normalized by the mean and standard deviation of the development set. For each caption in the dataset, we remove punctuation and convert all letters to lowercase to reduce the vocabulary size. To mark the beginning and the end of sentences, we add special tokens "<BOS>" and "<EOS>" to captions. The available training data is split into a model training part, consisting of 90% of available data and a held-out 10% validation set.

3.3. Evaluation metrics

A total of eight objective metrics are utilized to evaluate our model-generated captions: BLEU@1-4 grams [4], METEOR [7], Rouge-L [5], CIDEr [6] and SPICE [16]. A further SPIDEr metric is calculated as the mean of CIDEr and SPICE.

3.4. Training details

We submit predictions from four models to the challenge:

- CRNN-B (Base). This is our baseline CRNN-GRU encoder-decoder model.
- CRNN-W (Word). Here, the decoder word-embeddings are initialized from Word2Vec word-embeddings trained on the development set captions.
- CRNN-E (Ensemble). Here we fuse CRNN-B and CRNN-W results on output level.
- CRNN-R (Reinforcement). Here we finetune CRNN-W using reinforcement learning.

The details for each submission are elaborated in the following.

XE training For XE training, teacher forcing is used to accelerate the training process. We evaluate the model on the validation set at each epoch and select the best model according to the highest BLEU_4 score. We train the model for 20 epochs and use Adam [17] optimizer with an initial learning rate of 5×10^{-4}. The batch size is 32. According to whether Word2Vec is used for word embedding initialization, we get CRNN-B and CRNN-W respectively.

Ensemble In order to further enhance performance we merge the outputs of CRNN-B and CRNN-W on word-level. The encoded audio representation $\mathbf{v}$ is fed to both CRNN-B and CRNN-W to obtain two-word probabilities $\mathbf{p}_1$ and $\mathbf{p}_2$. We ensemble the two models, which means the current word is decoded according to the mean of $\mathbf{p}_1$ and $\mathbf{p}_2$. Then the current word embedding is fed to CRNN-B and CRNN-W to obtain the next word until <EOS> is generated.

Reinforcement The CRNN-R approach is first initialized by training a CRNN-W model using the standard XE criterion. This model is then finetuned using reinforcement learning, as seen in Section 2, by optimizing the CIDEr score using policy gradient with baseline normalization. Although [21] optimized SPIDEr by policy gradient in image captioning, we choose CIDEr as the training objective because CIDEr optimized model trained by SCST achieved better performance [10]. CIDEr measures sentence similarity through representation by n-gram TF-IDFs while BLEU focuses on "hard" n-gram overlaps. Such a "soft" similarity (CIDEr) may be a better optimization objective compared with BLEU under the condition that one audio corresponds to several semantic similar sentences, possibly composed of different n-grams. The model is trained for 25 epochs using Adam optimizer with a learning rate of 5×10^{-5}. Similar to the practice in XE training, we report the best model based on the CIDEr score on the validation set.

4. RESULTS

4.1. Results

Our results on the Clotho evaluation set are displayed in Table 1 and compared with the DCASE challenge baseline, which consists of a three-layer BiGRU encoder and two-layer BiGRU decoder. As it can be seen, our initial CRNN-B model largely outperforms the baseline, indicating that a potent encoder is indeed beneficial towards captioning performance. By initializing word embeddings with Word2Vec trained on the development set captions, CRNN-W gets a slight performance improvement in most metrics compared with CRNN-B, except CIDEr and METEOR. CRNN-E improves performance against both CRNN-B and CRNN-W. Our best performing model is CRNN-R. Interestingly, although CRNN-R is optimized towards CIDEr score, the relative improvement in BLEU_3 and BLEU_4 are more significant than CIDEr. The improvement in ROUGE_L and METEOR is not as significant as other metrics. However, CRNN-R does achieve the best performance in terms of all evaluation metrics, which validates the effectiveness of reinforcement learning for audio captioning.

With regards to the official challenge evaluation, our CRNN-R achieves the fourth place in DCASE2020 task 6 on the Clotho testing set. However, there is only a slight difference between our submission and the submission ranking the third (0.194 / 0.196).

Method	Model	$BLEU_1$	$BLEU_2$	$BLEU_3$	$BLEU_4$	$ROUGE_L$	CIDEr	METEOR	SPICE	SPIDEr
Baseline	GRU	0.389	0.136	0.055	0.015	0.262	0.074	0.084	0.033	0.054
1	CRNN-B	0.457	0.248	0.143	0.083	0.306	0.203	0.135	0.081	0.142
2	CRNN-W	0.459	0.253	0.151	0.086	0.314	0.192	0.133	0.083	0.138
3	CRNN-E	0.479	0.274	0.167	0.099	0.328	0.232	0.143	0.088	0.160
4	CRNN-R	0.529	0.335	0.226	0.146	0.352	0.280	0.149	0.099	0.194
Post-challenge	CRNN-R-SpecAug	**0.561**	**0.341**	**0.244**	**0.156**	**0.368**	0.338	**0.162**	**0.108**	0.223
Runner-up [18]	CNN-Transformer	0.532	0.341	0.227	0.149	0.354	**0.340**	0.157	0.108	**0.224**
Third [19]	CNN-LSTM	0.489	0.285	0.177	0.107	0.325	0.252	0.148	0.091	0.172

Table 1: Performance on the evaluation set of several submissions. (1) CRNN-GRU basline (2) with Word2Vec embedding intialization (3) ensemble of (1) and (2) (4) finetuning via reinforcement learning (CIDEr Loss). Post-challenge result: training of (4) with added data augmentation. Best performance is highlighted in bold. Winner results are not included since we only compare single model performance but the winner ensembles 20 models to get the result [20].

If we take a look at the overall metrics (excluding SPIDEr), our submission ranks second on four metrics ($BLEU_{1\text{-}3}$ and $ROUGE_L$), ranks third on three metrics ($BLEU_4$, METEOR and SPICE). Only for CIDEr, our submission ranks fourth. After the official challenge ended, we further experimented with data augmentation, namely spectrogram augmentation (SpecAug) [22]. Time masking and frequency masking are set with $T = 50, F = 10$. After adding SpecAug to our training strategy and retraining our CRNN-R for 200 epochs, the SPIDEr score is further boosted to 0.223, which is close to the submission ranking the second [18] (see Table 1).

It should be noted that our model is quite light-weight compared with other well-performing systems. While achieving a competitive result, our system only requires 5 Million parameters. By contrast, many other systems are in need of as many as 32 Million parameters (due to multiple models in ensemble) to achieve high performance.

4.2. Analysis

Generally, an encoder-decoder based architecture for audio captioning requires an effective encoder to embed audio into a single vector, and a decoder to generate an accurate description from the audio embedding. We incorporated CRNN which performs well in sound event detection (SED) as our encoder. Explicit training methods are explored in the top two submissions. [20] uses the multi-task learning mechanism, namely predicting caption and meta keywords, to train an encoder. [18] uses keywords extracted from the provided references to pretrain an encoder. To train a powerful decoder, we use reinforcement learning to optimize the evaluation metrics directly. Other submissions such as [20] add a sentence length prediction task to do multi-task training for the decoder, while [18] incorporates the transformer model [23] as their decoder. It can be concluded from the top submissions that information provided by meta or caption keywords are useful for encoder training with multi-task learning or explicit pretraining. In comparison, our proposed CRNN-GRU model using reinforcement learning is easy to train and does not require multiple label-types (e.g., keywords and metadata).

The relatively low score on CIDEr of our model may stem from CIDEr's characteristics. Compared with other metrics which focus on n-gram overlap, CIDEr calculates a sentence similarity based on term frequency-inverse document frequency (TF-IDF). Therefore it attaches more importance on low-frequency words. In our previous work, we find that our model tends to output general, repetitive sentences [3]. Even with reinforcement learning, optimizing CIDEr score in a mini-batch means that the estimated IDF can be inaccurate without access to the whole dataset reference captions. This exposure bias leads to bad CIDEr performance.

Lastly, we present an example of reference captions and our CRNN-R prediction. Even though our prediction accurately describes the audio event, it is not as detailed as the human annotations. The human annotations may contain specific descriptions like "vibrating""buzzing" while our model prediction only generates "running". Due to the limited information in audio as well as the direct optimization towards CIDEr metric, the model chooses to output a correct, yet general description of the audio events.

Example
Ref 1: a tractor or lawn mower runs its heavily vibrating engine
Ref 2: an engine or a machine of some sort running for the entirety
Ref 3: an engine or a machine runs along continuously
Ref 4: an engine with a heavy vibration coming from a tractor or lawn mower
Ref 5: a machine is buzzing and people are speaking in the background
Prediction: a machine is running while people are talking in the background

5. CONCLUSION

In this paper, we propose a novel audio captioning approach utilizing a CRNN encoder front-end as well as a reinforcement learning framework. Audio captioning models are trained on the Clotho dataset. The results on the Clotho evaluation set suggest that the CRNN encoder is crucial to extract useful audio embeddings for captioning while reinforcement learning further improves the performance significantly in terms of all metrics. Our approach ranked fourth in the DCASE2020 task 6 challenge testing set with a competitive result on all metrics except CIDEr. Notably, our approach is the best performing non-ensemble result without data augmentation, with the least parameters (5 Million). By further utilizing SpecAug data augmentation, we observe an additional boost in regards to the SPIDEr score on the evaluation set from 0.190 to 0.223.

6. ACKNOWLEDGMENT

Mengyue Wu and Kai Yu are the corresponding authors. This work has been supported by the Major Program of National Social Science Foundation of China (No.18ZDA293). Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

7. 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)*, Oct 2017, pp. 374–378.

[2] S. Lipping, K. Drossos, and T. Virtanen, "Crowdsourcing a dataset of audio captions," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1907.09238

[3] M. Wu, H. Dinkel, and K. Yu, "Audio Caption: Listen and Tell," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May. Institute of Electrical and Electronics Engineers Inc., may 2019, pp. 830–834.

[4] 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 on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[5] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," *Proceedings of the workshop on text summarization branches out (WAS 2004)*, 2004.

[6] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[7] 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.

[8] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[9] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057–1063.

[10] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 7008–7024.

[11] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, "An actor-critic algorithm for sequence prediction," *arXiv preprint arXiv:1607.07086*, 2016.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[13] H. Dinkel and K. Yu, "Duration Robust Weakly Supervised Sound Event Detection," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2020, pp. 311–315. [Online]. Available: https://ieeexplore.ieee.org/document/9053459/

[14] H. Dinkel, Y. Chen, M. Wu, and K. Yu, "GPVAD: Towards noise robust voice activity detection via weakly supervised sound event detection," mar 2020. [Online]. Available: http://arxiv.org/abs/2003.12222

[15] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *45th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Barcelona, Spain, May 2020. [Online]. Available: https://arxiv.org/abs/1910.09387

[16] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: semantic propositional image caption evaluation," *CoRR*, vol. abs/1607.08822, 2016. [Online]. Available: http://arxiv.org/abs/1607.08822

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

[18] 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., June 2020.

[19] H. Wang, B. Yang, Y. Zou, and D. Chong, "Automated audio captioning with temporal attention," DCASE2020 Challenge, Tech. Rep., June 2020.

[20] Y. Koizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "The NTT DCASE2020 challenge task 6 system: Automated audio captioning with keywords and sentence length estimation," DCASE2020 Challenge, Tech. Rep., June 2020.

[21] 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.

[22] 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," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[23] 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.