LEVERAGING STATE-OF-THE-ART ASR TECHNIQUES TO AUDIO CAPTIONING

Chaitanya Narisetty[1]*, Tomoki Hayashi*[2]*, Ryunosuke Ishizaki*[2]*,*
Shinji Watanabe[1]*, Kazuya Takeda*[2]

[1] Carnegie Mellon University, Pittsburgh, USA,
cnariset@andrew.cmu.edu, shinjiw@ieee.org
[2] Nagoya University, Nagoya, Japan,
{hayashi.tomoki, ishizaki.ryunosuke}@g.sp.m.is.nagoya-u.ac.jp,
takeda@is.nagoya-u.ac.jp

ABSTRACT

This paper details our work towards leveraging state-of-the-art ASR techniques for the task of automated audio captioning. Our model architecture comprises of a convolution-augmented Transformer (Conformer) encoder and a Transformer decoder to generate natural language descriptions of acoustic signals in an end-to-end manner. To overcome the limited availability of captioned audio samples for model training, we incorporate the Audioset-tags and audio-embeddings obtained from pretrained audio neural networks (PANNs) as an auxiliary input to our model. We train our model over audio samples from Clotho & AudioCaps datasets, and test over Clotho dataset's validation and evaluation splits. Experimental results indicate that our trained models significantly outperform the baseline system from DCASE 2021 challenge task 6.

Index Terms— Automated Audio Captioning, Conformer, ESPNet, PANNs

1. INTRODUCTION

Automated audio captioning was first proposed by [1] as a task of generating descriptive captions for a give audio signal using the concepts of audio processing and natural language processing. Datasets for this task consist of audio samples mapped to at least one corresponding human-generated caption [2, 3]. To generate a caption of sufficient quality, it is essential that the training model distills meaningful audible representations from an audio signal.

Similar to established image caption generators [4], a typical audio captioning model also comprises of an encoder-decoder framework. The encoder computes an encoded representation of relevant acoustic features in an input audio sample, and the decoder outputs a sequence of tokens using the encoded representation to form a suitable descriptive caption [1, 5]. Popular and effective frameworks for audio captioning in literature comprise of CNN encoders and Transformer decoders. A 10-layer CNN encoder and a Transformer decoder with multi-head self-attention was proposed by [5], where the CNN encoder was first pretrained for a multi-label classification task. To overcome the issue of limited number of training samples, [6] used a mix-up based data augmentation to create training samples from convex combinations of two given audio samples and their word token embeddings. Reinforcement learning in the form of self-critical sequence training (SCST), introduced for image captioning [7], was also explored for audio captioning by [8] to directly optimize the evaluation metrics (BLEU, CIDEr etc.) instead of the cross-entropy loss during greedy decoding at test-time.

Our proposed method is based on state-of-the-art automatic speech recognition (ASR) techniques such as convolution-augmented Transformer (Conformer) [9] and the fusion of a language model, incorporated in the end-to-end speech processing toolkit ESPnet [10]. Furthermore, we utilize the pretrained audio tagging model PANNs [11] to extract auxiliary information (e.g., Audioset [12] tags and embedding vector) and integrate them with the ASR model, enabling us to generate consistent captioning results. The contributions of this paper are as follows:

- We apply an attention-based encoder-decoder with the Conformer architecture, which allows capturing both local and global contexts in the input sequence. We also employ the language model trained on the captions and integrate its score with shallow fusion, resulting in a more stable prediction.
- We also introduce a pretrained audio tagging model PANNs to extract the auxiliary information, including Audioset tags and embedding vectors, and then utilize them as the additional inputs for the encoder-decoder model.
- Experimental evaluation with DCASE 2021 Task 6 dataset [13] shows that the proposed framework significantly outperforms the baseline system. Our best trained model shows a SPIDEr score of 0.224 and 0.246 on the development-validation and development-evaluation sets, respectively.
- This work expands on our DCASE2021 challenge report [14] with detailed description of the proposed framework and key insights into the contributions of auxiliary input features and language model fusion.
- Towards supporting accessible and reproducible research, we intend to release our audio captioning system and pretrained models to the ESPNet toolkit[1].

2. PROPOSED METHODOLOGY

2.1. Overview

Fig. 1 illustrates an overview of the proposed method. Similar to other speech-related tasks, we use log-mel filterbank features as the primary input. Data augmentation is performed over these

[1] https://github.com/chintu619/espnet/tree/aac_wordtokens/egs/clotho/aac_word

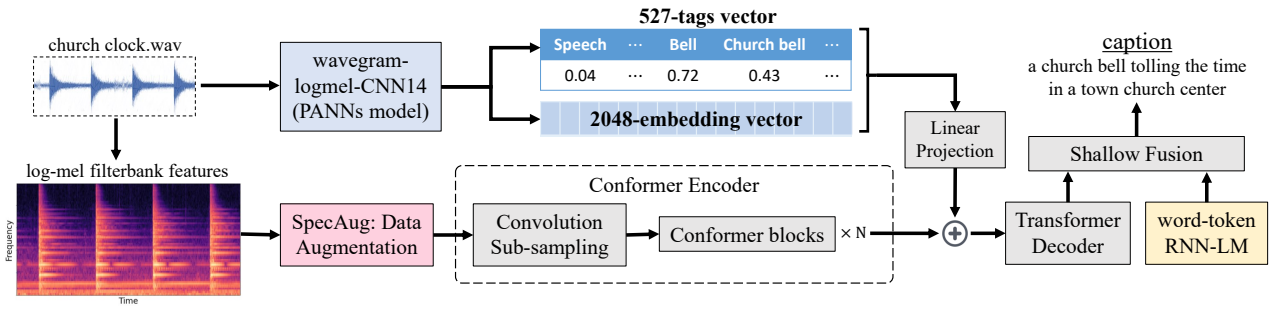

Figure 1: An overview of the proposed network architecture based on a Conformer encoder and a Transformer decoder. SpecAug based data augmentation is performed on the log-mel filterbank features. The pretrained wavegram-logmel-CNN14 PANNs model extracts the 527-tags vector and 2,048-embedding vector, and are fed as auxiliary inputs. Finally, a shallow fusion of decoder output and RNN-LM is performed to generate the output caption.

primary features to improve noise robustness. In addition to the primary input, we employ auxiliary inputs such as Audioset tags and an embedding vector, which are extracted with the pretrained audio tagging model PANNs [11]. Both inputs are fed into the attention-based encoder-decoder model. Inspired by the success of Conformer-based models for tasks like speech recognition, translation, and separation [15], our model uses a Conformer encoder for processing these audio features and a Transformer decoder to process words in a corresponding caption. To further improve the performance, we introduce the RNN-based language model and combine it with the encoder-decoder model in the decoding stage. The following subsections describe each of the components of our proposed Conformer model.

2.2. Encoder-Decoder Framework

The encoder incorporates a convolution sub-sampling layer and several Conformer blocks, where each block consists of a first feed-forward module (FFN), a multi-head self-attention (MHSA) module, a convolution module and a second feed-forward module in the aforementioned sequence. Similar to Transformer ASR models, a residual connection is added to the output of the feed-forward module followed by a layer normalization (LN) [16]. To regularize the network, the module employs dropout and Swish activation [17].

The self-attention module uses relative positional encoding in order to make the encoder robust to varying input length. This feature makes Conformer an ideal encoder for audio samples of varying length as seen in the present task. This module also employs dropout and a residual connection to regularize the network. For an input sequence $\mathbf{X} \in \mathbb{R}^{T \times d^{att}}$, where T is the number of time frames and d^{att} is the attention dimension, the positional encoding and regularization are computed according to Eq. 1. Finally the convolution module employs a point-wise convolution, a gated linear unit (GLU) activation [18], 1-dim depth-wise convolution layer, a batch normalization layer, Swish activation and a point-wise convolution. Both feed-forward modules employ a half-step scheme, and a residual connection and dropout are again used for regularization as shown in Eq. 2.

$$\mathbf{X} = \mathbf{X} + \text{Dropout}(\text{MHSA}(\text{LN}(\mathbf{X}))) \quad (1)$$

$$\mathbf{X} = \mathbf{X} + 0.5 \times \text{Dropout}(\text{FFN}(\text{LN}(\mathbf{X}))) \quad (2)$$

The decoder also incorporates several Transformer blocks, where each block consists of a multi-head self-attention layer, and a linear layer with ReLU activation sandwiched between two layer normalization layers.

2.3. Auxiliary Input Features

To improve the generalization ability of our model, we provide an auxiliary input to our encoder framework, similar to the use of robust audio embeddings in speaker recognition tasks [19]. For this purpose, we use CNN14 - one the PANNs models trained on the large scale Audioset dataset of over 5,000 hours of audio samples labeled with 527 audio tags. The CNN14 model is a wavegram-logmel-CNN system trained on 32kHz audio samples using 14 convolution layers. The model outputs a 527-tags vector, whose each element corresponds to the prediction of an audio tag. In addition to this 527-tags vector, we also extract a 2,048-embedding vector from each audio sample that is inputted to final classification layer.

The tags and/or embeddings obtained using PANNs are used as an auxiliary input to our model. When using both the tags and embeddings, the two feature vectors are simply concatenated to form a single column vector. These features are first L2 normalized and then passed through a feed-forward layer to be projected to the same size as that of the attention dimension. The projected features are finally added to the output of the Conformer encoder, before being sent as an input to the Transformer decoder.

2.4. Shallow Fusion with Language Model

We also separately train a word-token RNN language model (RNN-LM) using the captions in the training data and integrate it with the decoder using shallow fusion [20]. During inference, for each partial hypothesis h, the decoder combines its attention scores $\alpha_{att}(h)$ with the look-ahead word-token scores $\alpha_{lm}(h)$ provided by RNN-LM according to Eq. 3, where γ is a scaling factor.

$$\alpha(h) = \alpha_{att}(h) + \gamma \cdot \alpha_{lm}(h) \quad (3)$$

3. EXPERIMENTS

3.1. Data Preparation and Pre-processing

Our proposed model takes 16 kHz audio samples as input and computes 80 log-mel energies from each 64 ms frame, shifted every 32 ms. Accordingly, all the audio files in Clotho-v2 dataset were downsampled from 44.1 kHz to 16 kHz. The overall development split of the Clotho-v2 dataset has 3,839 training samples, 1,045 validation samples and 1,045 evaluation samples. Each audio sample is 15-30

seconds long and contains 5 human generated captions with 8-20 words each. Since the Clotho-v2 dataset is relatively small to train large neural networks, we additionally augment the training data with roughly 46,000 single caption audio samples from the AudioCaps dataset [3]. Audio samples in this dataset are carefully chosen from the 2M samples in Audioset dataset [12]. Each audio sample is roughly 10 seconds long.

We perform input feature augmentation using SpecAug [21] consisting of three kinds of deformations - time warping, frequency masking and time masking. We set the maximum time warp parameter to $W = 5$, and randomly choose $w \in [0, W]$ such that the log-mel filterbank feature matrix is warped by w. Frequency and time masking are based on Cutout [22] regularization technique which masks a randomly chosen rectangular portion of the log-mel filterbank matrix. Dimensions of the mask were chosen randomly based on the maximum frequency and time masking parameters of $F_m = 30$ and $T_m = 40$ respectively.

3.2. Comparison Models and Training Parameters

3.2.1. Baseline System

The DCASE 2021 challenge task 6 provided a baseline encoder-decoder framework consisting of a 3 layer bi-directional GRU encoder, and a decoder with one GRU layer and one classification layer. The input acoustic features are extracted using 64 log-mel energies estimated over a 46ms frame, shifted every 23ms. Each encoder and decoder GRU layer has 256 bi-directional features, and the classification layer outputs the probability of 4637 unique words in each decoder iteration (time-step).

3.2.2. Proposed Model

The proposed Conformer model used in our experiments has 16 encoder layers and 4 decoder layers, each with 1,024 units along with 4 heads, $d^{att} = 256$ for attention layers and a depth-wise convolution with kernel size of 15. For better predictive performance through model ensemble, we also explored several variations in the dimensions of the proposed model. Decoding module for a model ensemble performs posterior averaging of the attention score output of constituent model decoders. A variation of the proposed Conformer model was trained with smaller encoder-decoder layers having 512 units each. Another variation was trained with a smaller attention framework having 128-dim layers with 2 heads. Final model variation was trained with above mentioned smaller attention framework, but with a larger kernel size of 31.

In addition to the log-mel energies, we extract a softmax vector of 527-tags and a 2,048-embedding vector from each audio sample using the CNN14 PANNs model [11] as detailed in Section 2.3. Each element of 527-tags vector represents the probability of a corresponding class-label in the Audioset ontology. All the proposed model variations employ shallow fusion using a 2-layer RNN-LM trained for 25 epochs with a batch-size of 64 and dropout of 0.5. Scaling factor γ for shallow fusion is set to 0.2.

3.2.3. Hyper-Parameters

During training, 64 audio-caption pairs were batched together and trained for 50 epochs with a learning-rate of 0.5, dropout of 0.1, cross-entropy loss function and *noam* optimizer [23]. To prevent exploding gradients, we set the gradient threshold to 5. Label smoothing [24] was set to 0.1 to avoid high confidence training predictions. Upon completion of training, we average the model parameters over the final-10 epochs and this averaged model was used for inference. During inference, beam search was performed with a beam-size of 10 and RNN-based language model weight of 0.2. We note that the above hyper-parameters are optimized based on our prior experience in tuning ASR systems.

3.3. Evaluation Metrics

Experimental evaluation for audio captioning is conducted using six metrics: BLEU-n [25], ROUGE-L [26], METEOR [27], CIDEr [28], SPICE [29] and SPIDEr [30]. Precision of output captions for 1,2,3,4-grams (contiguous sequence of n words) are evaluated by BLEU-n. F-measure between output and ground-truth captions is estimated by ROUGE-L by estimating their longest common subsequence. METEOR is a machine translation metric which computes a harmonic mean of 1-gram precision and recall between output and ground-truth captions. CIDEr computes the average cosine similarity of n-grams between output and ground-truth captions. SPICE evaluates the semantic similarity between output and ground-truth captions by first performing lemmatisation of captions and then computing the F-score between their scene graphs. Lemmatisation maps all the inflected forms of a word to its root form, and scene graphs are a semantic representation which encode the objects, attributes and relations present in captions. SPIDEr simply computes an average score of CIDEr and SPICE metrics.

4. RESULTS

The performance of our trained models were evaluated on both the development-validation and development-evaluation splits and are summarized in Table 1 and Table 2. All our proposed models outperform the DCASE 2021 baseline system by a significantly margin. Summarized results also show the contribution from various components of our proposed model: encoder-decoder, self-attention and auxiliary features.

4.1. Observations

We observe a slight degradation in performance when varying our model's architecture as compared to the baseline Conformer model. However these variations help to improve the performance of a model ensemble. Auxiliary input features of tags and embeddings were able to improve the scores of most metrics, especially over the development-validation split. We also observe that augmenting the training data with the development-evaluation split was indeed able to improve the proposed Conformer's performance over the development-validation split and vice-versa. Model ensemble was also performed over various combinations of our trained models, and was further able to increase the overall system performance.

4.2. Discussion

4.2.1. Understanding Auxiliary Features

We explore the individual contribution of extracted tags and embeddings towards the performance boost provided by the auxiliary input features. Table 3 details the performance of the proposed Conformer model when trained with only the extracted 2,048-embeddings and 527-tags as secondary inputs. Although the extracted tags provide sufficiently good CIDEr score, using both the tags and embeddings improves the SPICE score.

We additionally observed that the captions generated using Conformer model with auxiliary features for 520 samples ($\sim 25\%$), among the combined 2090 validation and evaluation samples, had

Method	BLEU-1,2,3,4				ROUGE-L	METEOR	CIDEr	SPICE	SPIDEr
Baseline	0.389	0.136	0.055	0.015	0.262	0.074	0.084	0.033	0.054
Conformer	0.512	0.317	0.205	0.131	0.336	0.148	0.310	0.100	0.205
smaller enc-dec	0.500	0.311	0.203	0.129	0.336	0.144	0.299	0.099	0.199
smaller attention	0.490	0.307	0.199	0.127	0.332	0.143	0.310	0.096	0.203
+ larger-kernel	0.496	0.307	0.198	0.124	0.336	0.143	0.297	0.098	0.198
+ auxiliary features	**0.521**	**0.330**	**0.217**	**0.138**	**0.345**	**0.154**	**0.323**	**0.107**	**0.215**
+ dev-eval split	0.515	0.321	0.207	0.131	0.340	0.149	0.314	0.101	0.208
Ensemble	0.533	0.343	0.226	0.146	0.355	0.154	0.341	0.106	0.224

Table 1: Scores of evaluation metrics for the development-validation split.

Method	BLEU-1,2,3,4				ROUGE-L	METEOR	CIDEr	SPICE	SPIDEr
Baseline	0.378	0.119	0.050	0.017	0.078	0.263	0.075	0.028	0.051
Conformer	0.534	0.343	**0.233**	**0.158**	0.354	0.157	0.351	0.106	0.228
smaller enc-dec	0.524	0.331	0.219	0.144	0.356	0.153	0.329	0.103	0.216
smaller attention	0.506	0.320	0.212	0.140	0.349	0.152	0.337	0.102	0.219
+ larger-kernel	0.518	0.330	0.224	0.150	0.355	0.154	0.340	0.105	0.223
+ auxiliary features	0.536	0.341	0.225	0.146	0.357	0.160	0.346	0.108	0.227
+ dev-val split	**0.541**	**0.346**	0.231	0.152	**0.356**	**0.161**	**0.362**	**0.110**	**0.236**
Ensemble	0.546	0.356	0.243	0.165	0.369	0.163	0.381	0.110	0.246

Table 2: Scores of evaluation metrics for the development-evaluation split.

a SPICE score of zero. Note that SPICE score is measured over all 5 ground-truth captions and these zero scores can imply a complete semantic mismatch for a significant portion of testing samples. Among these zero score samples, we also observe that extracted AudioSet tags (auxiliary features) are sometimes match very closely with the caption words. Consider '`18 Little Group.wav`', an audio sample from validation split with a ground-truth caption of '*sea animals make strange blips, groans and other vocalizations*'. Our generated caption is '*a cat is meowing and making noises*'. However, the top-2 AudioSet tags extracted for this audio sample are '*Whale vocalization*' and '*Animal*'. A potential improvement from this analysis would be to increase the weight of projected auxiliary features when mixing them with the encoder output. To better integrate the extracted tags and embeddings, it is also possible to use an additional pretrained encoder from the PANNs model, and fine-tune the auxiliary features during training.

Method	CIDEr	SPICE	SPIDEr
Conformer + auxiliary input	0.323	**0.107**	**0.215**
- 527-tags	**0.325**	0.102	0.214
- 2048-embeddings	0.315	0.098	0.207
Conformer + auxiliary input	**0.346**	**0.109**	**0.227**
- 527-tags	**0.346**	0.104	0.225
- 2048-embeddings	0.342	0.106	0.224

Table 3: Evaluating contributions of PANNs tags and embeddings towards model performance on development-validation split (top) and development-evaluation split (bottom).

4.2.2. Evaluating Shallow Fusion with RNN-LM

Shallow fusion with a pretrained language model is equivalent to a model ensemble approach where the scores of the acoustic model and the language model are combined. Table 4 shows the performance improvement, especially of CIDEr scores, provided by an RNN-LM optimized on the word sequences in the training dataset.

Method	CIDEr	SPICE	SPIDEr
Conformer	0.310	0.100	0.205
- RNN-LM	0.300	0.098	0.199
Conformer	0.351	0.106	0.228
- RNN-LM	0.344	0.105	0.225

Table 4: Evaluating contribution of RNN-LM towards model performance on development-validation split (top) and development-evaluation split (bottom).

5. CONCLUSION

This work provides a detailed description and analysis of our submission to 2021 DCASE challenge Task 6: automated audio captioning. The proposed methodology employs existing state-of-the-art ASR techniques including Conformer-encoder, Transformer-decoder, data augmentation, Audioset tags & embeddings as auxiliary inputs and shallow fusion with a pretrained RNN language model. Our experiments qualify the ability of ASR techniques for effective captioning of audio samples by significantly outperforming the DCASE baseline system. Leveraging ASR techniques for audio captioning opens potential research directions towards developing an integrated framework for joint modeling of ASR and captioning tasks, and will be tackled as part of our future work.

6. ACKNOWLEDGMENT

This work was supported in part by Sony Corporation, JHU HLTCOE, and Bridges PSC (TG-CIS210014).

7. REFERENCES

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

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

[3] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *NAACL-HLT*, 2019.

[4] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3156–3164.

[5] 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 *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020). Tokyo, Japan*, 2020, pp. 21–25.

[6] 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.

[7] 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.

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

[9] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.

[10] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211.

[11] 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.

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

[13] 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.

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

[15] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi, *et al.*, "Recent developments on espnet toolkit boosted by conformer," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[16] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[17] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[18] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*. PMLR, 2017, pp. 933–941.

[19] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[20] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Proceeding of Interspeech*, 2017, pp. 949–953.

[21] 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.

[22] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[24] 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.

[25] 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.

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

[27] A. Lavie and A. Agarwal, "Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proceedings of the second workshop on statistical machine translation*, 2007, pp. 228–231.

[28] 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.

[29] 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.

[30] 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.