IMPROVING AUTOMATED AUDIO CAPTIONING FLUENCY THROUGH DATA AUGMENTATION AND ENSEMBLE SELECTION

Jaewon Kim[*]*, Yoon-Ah Park*[*]*, Jae-Heung Cho*[*]*, and Joon-Hyuk Chang*

Department of Electronic Engineering, Hanyang University,
Seoul, Republic of Korea

ABSTRACT

Automated audio captioning is a task of generating descriptions corresponding to audio clips. The training process of AAC typically consists of a pre-training, fine-tuning, and reinforcement learning. While reinforcement learning enhances the evaluation metrics for captions, it has the drawback of potentially lowering the quality of the captions, such as incomplete sentence or repetitive words. In this study, we propose an ensemble selection technique that combines models before and after reinforcement learning to improve evaluation metrics while maintaining caption quality. Furthermore, we apply several data augmentation techniques to complement the characteristics of WavCaps, which predominantly consists of single events, and improve generalization property. In particular, proposed approaches can reach impressive scores both an existing metric SPIDE$_r$, and a new fluency metric SPIDE$_r$-FL, 0.344 and 0.315, respectively. This resulted in a 2nd place ranking in DCASE 2023 task 6a, while the baseline system achieved SPIDE$_r$ of 0.271 and SPIDE$_r$-FL of 0.264.

Index Terms— Automated audio captioning, pre-training, data augmentaion, reinforcement learning

1. INTRODUCTION

Automated audio captioning (AAC) is an audio-to-text generation task that first introduced by K. Drossos *et al*. [1]. It is an multimodal task combines audio processing and natural language processing to describe audio clips using natural language. Unlike sound event detection [2] and audio classification tasks [3], AAC aims to capture spatio-temporal relationships in audio clips and perform advanced interpretation of audio. The detection and classification of acoustic scenes and events (DCASE) challenge has played a significant role in promoting research on AAC, particularly with the use of audio-caption pair datasets like Clotho [4] and AudioCaps [5].

During the initial development of AAC models, recurrent neural network (RNN)-based approaches [1, 6, 7] were commonly proposed. Moreover, as attention-mechanism language models [8] with superior performance emerged, transformer-based models gained significant popularity. Various transformer-based architectures, including convolution neural network (CNN)-transformer [9, 10], transformer [11], and CNN-RNN-transformer [12] with encoder-decoder structures, were widely adopted. These models establish a crucial connection between audio and transformer-based language models. CNN-based encoders have particularly demonstrated outstanding performance in audio representation as audio feature extractors. This combination of transformers and CNN-based encoders has significantly advanced the field of AAC.

In this study, we employ a bidirectional auto-regressive transformer (BART) [13] based CNN-BART model. In addition, we used data augmentation techniques such as SpecAugment [14], PairMix [15], and synonym substitution in the pre-training and fine-tuning process to enhance the generalization characteristics of the model and complement the characteristics of the dataset. SpecAugment is a widely used technique that applies random transformations to the log mel-spectrogram of the audio input, thereby enhancing robustness and generalization. PairMix is a multimodal data augmentation technique that mixes two audio clips and captions. The WavCaps [16] we used in the pre-training process mostly consisted of single event audio clips; therefore, model could not be sufficiently training about the spatial-temporal features. To address these issues, we used PairMix in the pre-training phase. Additionally, to enhance the model's universality and prevent overfitting during fine-tuning, we conducted synonym substitution, which entailed replacing random words with their synonym within the caption.

Reinforcement learning (RL) was adopted to futher enhance the model's performance. Specifically, we utilized RL based on self-critical sequence training, which has been proposed as a supplementary method to directly improve evaluation metrics. Throughout the RL process, we monitored the CIDE$_r$ [17] score, resulting in significant improvements in SPIDE$_r$. However, it is worth noting that RL models often generate captions of lower quality, such as incomplete sentences or repetitive words, as their primary objective is to improve the CIDE$_r$ score. In this study, we proposed an ensemble selection technique that can maintain the advantages of RL while enhancing caption quality. By combining models trained without RL and models trained with RL, we observed improvements in both SPIDE$_r$ and SPIDE$_r$-FL scores compared to using the pre-trained model alone. Also, the proposed method showed higher performance than the existing models in terms of SPIDE$_r$ and SPIDE$_r$-FL.

2. RELATED WORKS

AAC task employs various data augmentation techniques to enhance model performance and improve generalization capabilities. These techniques include SpecAugment, mix up, time stretching, white noise injection, and more. Among them, SpecAugment is widely used as a key data augmentation technique in AAC. It involves transforming spectrogram data in the frequency domain to increase data diversity. Frequency domain transformations can be performed in various ways, such as time masking, where a portion of the time axis is masked or duplicated, and frequency masking, where certain frequency ranges are masked. These transformations

[*]: Equal contributions.

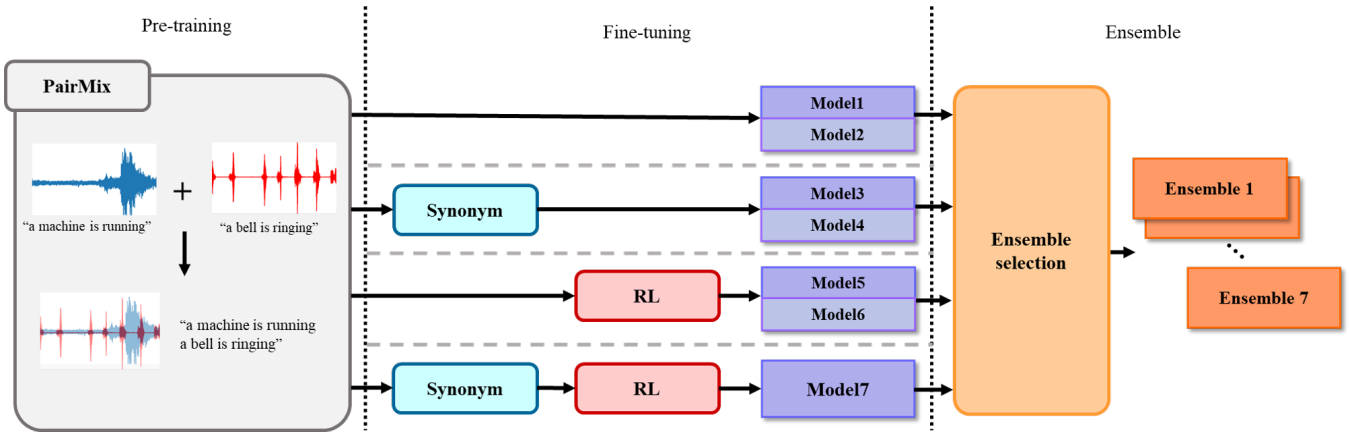

Figure 1: The flow of overall system. Synonym and RL refer synonym substitution and reinforcement learning.

allow the model to learn from a wider range of frequency patterns and enhance its robustness to noise and environmental changes, thereby improving both performance and generalization abilities. SpecAugment is recognized as a vital component of data augmentation and is widely adopted in research and applications. By utilizing this technique, AAC models can effectively operate in various environments and noise conditions. Additionally, mix up [18] is a technique where two audio samples are linearly combined to create a new sample. This allows the model to learn features from different audio sources and diversify the training data. PairMix linearly combines two captions to generate a new caption, which helps improve the model's learning by creating diverse combinations of sentence structures and content.

Captioning models are usually trained using cross-entropy loss. However, it should be noted that minimizing the loss function does not always improve the evaluation metric. To address these challenges, we employ a technique called self-critical sequence training [19]. This approach allows us to optimize the evaluation metrics directly, leading to improved scores in terms of these metrics. The model generates captions, and rewards are computed based on the metrics (such as SPICE [20], BLEU, CIDE$_r$) between the generated captions and the ground truth captions. The model is trained to maximize these rewards, aiming to generate superior captions.

3. METHODS

3.1. Data augmentation

3.1.1. PairMix

PairMix is an efficient and straightforward multimodal data augmentation technique in AAC task. It is first introduced in image captioning field named MixGen [21]. PairMix combines two audio clips and concatenates their corresponding captions. The formula for this process can be represented as follows:

$$\hat{\mathbf{a}} = \sum_{i=1}^{N} \lambda_i \mathbf{a}_i, \tag{1}$$

$$\hat{\mathbf{t}} = Concat(\mathbf{t}_{i=1}^{N}), \tag{2}$$

where $\mathbf{a}$, $\mathbf{t}$, $\hat{\mathbf{a}}$, and $\hat{\mathbf{t}}$ represent the audio waveform, caption, augmented audio, and augmented caption, respectively. $\lambda_i \in [0, 1]$ for i = 1, 2, ..., N is a hyperparameter that controls the degree of mixing.

Although data augmentation in the multimodal domain often poses challenges, PairMix provides an uncomplicated solution for audio-text datasets. By merging two audio clips, the model trained using PairMix data augmentation can extract multiple simultaneous sound events. This capability is crucial in AAC because detecting multiple sound events significantly improves the accuracy of the resulting captions. Simultaneously, the concatenation of two captions provides the model with the potential to generate more detailed and extended descriptions of audio clips. Hence, PairMix effectively enhances both audio feature extraction of detecting multiple sound events and caption quality of generating longer, specific descriptions.

3.1.2. Synonym substitution

Synonym substitution is a simple but effective data augmentation technique [22] derived from WordNet-based synonym substitution. This method entails substituting certain words in a sentence with their synonyms, thus allowing the model to express audio clips using rich vocabulary. During the fine-tuning process, we select individual words from the target captions, particularly nouns, and replaced them with their synonyms at random. This strategy can improve the generalization property of the model and the semantic properties of generated captions by ensuring that a single audio clip does not correspond to a single caption, but to various captions with the same meaning.

3.2. Audio feature extractor

In our model, we employ a 14-layer CNN derived from the pre-trained audio neural networks (PANNs) [23] architecture for the extraction of audio features. The choice of PANNs as an audio feature extractor through transfer learning is both rational and effective, given its pre-training on an audio tagging dataset. Audio tagging involves a multi-label classification task, necessitating the model to identify overlapping events occurring simultaneously within an audio clip. This requirement aligns well with the AAC task, which also needs to discern overlapping sound events. This particular CNN architecture is acknowledged for its great performance in capturing audio representations. It comprises six convo-

lutional blocks, each containing two CNN layers with a kernel size of 3×3. Following each CNN layer, batch normalization [24] is used to standardize the inputs, and a rectified linear unit (ReLU) activation function [25] is incorporated to enhance performance.

3.3. Language model

We incorporated BART as our language model, motivated by its impressive track record in text generation tasks. BART comprises an encoder and a decoder, each constructed from 12 transformer layers. The BART encoder receives the audio features produced by the audio feature extractor. In contrast, the BART decoder ingests both the output of the BART encoder and the reference caption. An attention mechanism is employed between the BART encoder and decoder, facilitating the model in capturing the semantic nuances and contextual information within the input sentence. Within each transformer block of the decoder, self-attention is applied to model the interactions among all the words in the input sentence. This strategy enables the model to generate precise predictions for the subsequent word, leading to high-quality text generation. The application of self-attention aids the model in capturing long-range dependencies and complex contextual relationships between words.

3.4. Ensemble selection

When choosing models for an ensemble, the conventional approach is to select those that perform well on particular evaluation metric. In the context of AAC, one of CIDE$_r$ or SPIDE$_r$-FL is often considered when forming ensemble combinations. However, this can lead to an imbalance, where one metric's increses while the other remains unchanged or decrease. This situation is particularly evident when there's a large difference between CIDE$_r$ and SPIDE$_r$-FL scores, often occurring when the model is trained using RL. The RL method, SCST, specifically targets the CIDE$_r$ evaluation metric score. While this approach elevates the CIDE$_r$ score, it tends to lower the SPIDE$_r$-FL score. In order to simultaneously boost both scores, we strategically select the models for the ensemble. Some of these models are already trained using RL, while others are not. Given that CIDE$_r$ score can be elevated sufficiently due to RL, we exclude models achieve low scores on SPIDE$_r$-FL for attaining greater scores of it. This method aims to ensure high performance on both the CIDE$_r$ and SPIDE$_r$-FL metrics. We will describe about the metrics in subsection 4.4.

4. EXPERIMENTS

4.1. Training

Our learning process consists of three stages: pre-training, fine-tuning and ensemble selection. During the pre-training phase, we employed the WavCaps, AudioCaps, and Clotho datasets to train the model, integrating the PairMix augmentation technique. Subsequently, in the fine-tuning phase, we froze the audio feature extractor and fine-tuned the model using the Clotho dataset via various methods. Some of the experiments utilized data augmentation techniques, while others did not. Similarly, a subset of the models was fine-tuned using a RL approach, while others were not. In the final phase, we created several combinations of the outcomes from the fine-tuning step to form an ensemble. Fig. 1 shows the overview of our proposed methods.

4.2. Dataset

4.2.1. WavCaps

The WavCaps dataset[1] is a large-scale, weakly-labelled audio captioning dataset, encompassing approximately 400,000 audio clips paired with captions. This dataset is including BBC Sound Effects, FreeSound [26], SoundBible and AudioSet [27]. To reduce the challenges associated with noisy and unsuitable raw descriptions, a three-stage processing pipeline leveraging ChatGPT is employed. The average duration of the audio clips is 67.59 seconds, and captions primarily consist of single-event descriptions, with an average caption length of 7.8 tokens. However, due to the unavailability of some data from FreeSound, we focused exclusively on the publicly accessible data for our research.

4.2.2. AudioCaps

AudioCaps is a dataset composed of 46,000 audio clips, each 10 seconds in duration and paired with text descriptions. The dataset is divided into three subsets: development-training, development-validation, and development-testing, which contain 38,118, 500, and 979 audio clips, respectively. While the training set provides a single caption per audio clip, the validation and testing sets offer five captions for each clip.

4.2.3. Clotho

Clotho v2.1 is divided into three subsets within its published development sets: development-training, development-validation, and development-testing. The development-training subset comprises 3,839 audio clips, and the development-validation and development-testing subsets each consist of 1,045 audio clips. All audio files in this dataset fall within a duration of 15 to 30 seconds. For each audio clip, there are five accompanying captions, each ranging from 8 to 20 words in length.

4.3. Experiment setup

The proposed model was trained using Adam [28] optimizer with batch size of 16 in both pre-training and fine-tuning phases. In pre-training phase, the learning rate was fixed to 1×10^{-6}, and in fine-tuning phases, we used two different learning rates of 5×10^{-5} and 1×10^{-6}. We adopted PairMix technique during pre-training process and we set $\lambda = 0.5$ and $N = 2$ in Eq. (1) and Eq. (2). With regard to synonym substitution, we randomly selected 8 captions in mini batch and substituted one nouns to another similar meaning nouns. In terms of ensemble selection, we selected models as following rules. First, exclude two models attaining the lowest and second lowest scores on SPIDE$_r$ (PairMix 1, PairMix 2) and also SPIDE$_r$-FL (PairMix+RL 1, PairMix+S·S+RL), respectively. Second, exclude two models attaining the lowest scores on SPIDE$_r$ and SPIDE$_r$-FL (PairMix 1, PairMix+RL 1). Finally, exclude none of them.

4.4. Evaluation metrics

We evaluated the models trained by our methods through one machine translation metric, METEOR [29], and four captioning metrics. CIDE$_r$, SPICE, SPIDE$_r$ and SPIDE$_r$-FL are those. METEOR assesses translation quality through exact word matches,

[1] https://github.com/XinhaoMei/WavCaps

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL
PairMix 1	*0.179*	*0.458*	*0.125*	*0.291*	0.290
PairMix 2	0.183	0.468	0.130	0.299	0.295
PairMix+S·S 1	0.182	0.473	0.129	0.301	0.298
PairMix+S·S 2	0.188	0.483	0.137	0.310	**0.306**
PairMix+RL 1	0.192	0.505	0.135	0.320	*0.154*
PairMix+RL 2	0.193	0.518	0.142	0.330	0.227
PairMix+S·S+RL	**0.195**	**0.526**	**0.143**	**0.335**	0.226

Table 1: Performances of each data augmentation techniques and RL on Clotho evaluation split. For all metrics, higher values indicate better performance. S·S refers synonym substitution. The difference between the number of models is the learning rate. Models possessing number 1 in their names are trained with learning rate of 5×10^{-5} and the others are 1×10^{-6}.

stem matches, synonym matches and phrase matches. Then it computes the harmonic mean of precision and recall according those matches. CIDE$_r$ measures weighted sum of cosine similarity between predicted and reference captions by term frequency and inverse document frequency so that it shows how created caption is well related to audio clip. SPICE metric calculates F-score using semantic scene graphs in sense of words relations in the captions. This means SPICE score can indicate model ability to generate semantically correct captions. SPIDE$_r$ is the average of CIDE$_r$ score and SPICE score, which is able to estimate the balance between two metrics. SPIDE$_r$-FL is an evaluation metrics that includes the fluency of captions. It is calculated by dividing the SPIDE$_r$ score by 10 for each individual example with an error.

5. RESULTS

The results of data augmentation techniques and RL on Clotho test set are shown in Table 1. We observed synonym substitution slightly enhances both SPIDE$_r$ and SPIDE$_r$-FL scores. Additionally, we compared the models trained with RL and those that are not. The models trained with RL were scoring higher values of SPIDE$_r$ than those models without, however, one of the captions of the highest SPIDE$_r$ score model was *'a fishing line is being wound up and a keys in'* which was not fluent enough since the sentence was not terminated. This results in the SPIDE$_r$-FL scores were significantly lower than the models not trained with RL. As a result, the model trained with PairMix and synonym substitution with learning rate 1×10^{-6} appeared the highest score of SPIDE$_r$-FL. Meanwhile, the model trained with PairMix, synonym substitution and RL was seem to be the top SPIDE$_r$ score model. In the context of ensemble selection, we analyzed the relations of ensembles with and without RL. The ensemble model excluding the RL scored similar in both SPIDE$_r$ and SPIDE$_r$-FL metrics with the top SPIDE$_r$-FL single model. However, when at least one model trained through RL was included in the ensemble, there was a notable increase in SPIDE$_r$ scores. Furthermore, for SPIDE$_r$-FL, some of these models achieved higher scores compared to ensembles without RL. Especially ensemble 3 model was achieving the highest score on SPIDE$_r$-FL metric. We also observed the caption improvement like following with the same audio clip we stated above: *'a fishing reel is being wound up and a bell is ringing'*. This caption is clearly more fluent. From Fig. 2, we noticed 4:1 ratio of non-RL models and RL models was performing the best. The model combination of each ensemble model is described below.

Model	METEOR	CIDEr	SPICE	SPIDEr	SPIDEr-FL	# of model
Ensemble 1	0.185	0.485	0.132	0.308	0.305	4
Ensemble 2	0.196	0.537	0.144	0.341	*0.256*	5
Ensemble 3	0.195	0.539	0.144	0.341	**0.332**	5
Ensemble 4	0.195	*0.529*	0.144	0.336	0.279	5
Ensemble 5	0.196	**0.543**	0.146	**0.345**	0.277	6
Ensemble 6	0.195	0.535	0.145	0.340	0.311	6
Ensemble 7	0.196	0.542	**0.147**	0.344	0.298	7

Table 2: Results of ensemble selection.

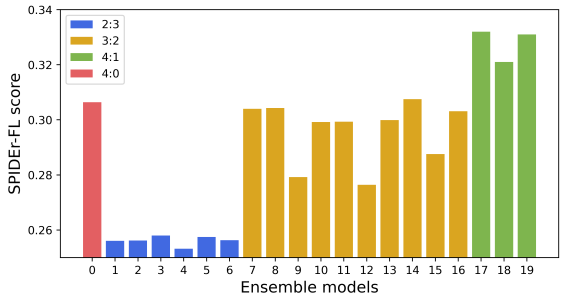

Figure 2: Ensemble SPIDE$_r$-FL scores according to ratio of non-RL and RL models. The legend shows the ratio according to colors.

- **Ensemble 1**: 4 models trained without RL.
- **Ensemble 2**: Top 5 SPIDE$_r$ models.
- **Ensemble 3**: Top 5 SPIDE$_r$-FL models.
- **Ensemble 4**: Excluding the lowest SPIDE$_r$ model and SPIDE$_r$-FL model.
- **Ensemble 5**: Top 6 SPIDE$_r$ models.
- **Ensemble 6**: Top 6 SPIDE$_r$-FL models.
- **Ensemble 7**: All 7 models.

6. CONCLUSION

In this study, we presented data augmentation, RL, and ensemble selection to boost both evaluation metrics, SPIDE$_r$ and SPIDE$_r$-FL. PairMix successfully rose the performance during the pre-training phase. This was considered the result of PairMix effect of developing the ability to detect multiple sound events at the same time stamps. Synonym substitution, likewise, elevated the model capability to express in various vocabulary. In terms of RL, it only concentrated on increasing the value of metric score, the actual fluency of captions decrease. This led to conclude removing RL models for ensemble was reasonable choice, however, those ensemble models including RL models were showing better performance when they were evaluated with both SPIDE$_r$ and SPIDE$_r$-FL. The chosen RL models played a role of regularization on ensemble, leading to generate well-related and more fluent captions.

7. ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2023-002255630, Development of Artificial Intelligence for Text-based 3D Movie Generation)

8. REFERENCES

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

[2] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.

[3] D. Stowell *et al.*, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[4] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2020, pp. 736–740.

[5] C. D. Kim *et al.*, "Audiocaps: Generating captions for audios in the wild," in *Proc. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 119–132.

[6] X. Xu *et al.*, "Audio caption in a car setting with a sentence-level loss," in *Proc. International Symposium on Chinese Spoken Language Processing* (*ISCSLP*), 2021, pp. 1–5.

[7] M. Wu *et al.*, "Audio caption: Listen and tell," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2019, pp. 830–834.

[8] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[9] K. Chen *et al.*, "Audio captioning based on transformer and pre-trained cnn." in *Proc. Detection and Classification of Acoustic Scenes and Events* (*DCASE*), 2020, pp. 21–25.

[10] X. Mei *et al.*, "Diverse audio captioning via adversarial training," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2022, pp. 8882–8886.

[11] X. Mei *et al.*, "Audio captioning transformer," *arXiv preprint arXiv:2107.09817*, 2021.

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

[13] M. Lewis *et al.*, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.

[14] D. S. Park *et al.*, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[15] E. Kim *et al.*, "Improving audio-language learning with mixgen and multi-level test-time augmentation," *arXiv preprint arXiv:2210.17143*, 2022.

[16] X. Mei *et al.*, "WavCaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research," *arXiv preprint arXiv:2303.17395*, 2023.

[17] R. Vedantam, Z. C. Lawrence, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2015, pp. 4566–4575.

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

[19] S. J. Rennie *et al.*, "Self-critical sequence training for image captioning," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2017, pp. 7008–7024.

[20] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Proc. 14th European Conference Computer Vision*, 2016, pp. 382–398.

[21] X. Hao *et al.*, "Mixgen: A new multi-modal data augmentation," in *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 379–389.

[22] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[23] Q. Kong *et al.*, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition." *IEEE Trans. Audio, Speech, and Language Processing.*, vol. 28, pp. 2880–2894, 2020.

[24] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. International Conference on Machine Learning*, 2015, pp. 448–456.

[25] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[26] F. Font *et al.*, "Freesound technical demo," in *Proc. International Conference on Multimedia*, 2013, pp. 411–412.

[27] J. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing* (*ICASSP*), 2017, pp. 776–780.

[28] Kingma *et al.*, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Agarwal and A. Lavie, "METEOR: An automatic metric for mt evaluation with high levels of correlation with human judgments," in *Proc. Workshop on Machine Translation*, 2007.