EFFECTS OF WORD-FREQUENCY BASED PRE- AND POST- PROCESSINGS FOR AUDIO CAPTIONING

Daiki Takeuchi, Yuma Koizumi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino

NTT Corporation, Japan

ABSTRACT

The system we used for Task 6 (Automated Audio Captioning) of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020 Challenge combines three elements, namely, data augmentation, multi-task learning, and post-processing, for audio captioning. The system received the highest evaluation scores, but which of the individual elements most fully contributed to its performance has not yet been clarified. Here, to asses their contributions, we first conducted an element-wise ablation study on our system to estimate to what extent each element is effective. We then conducted a detailed module-wise ablation study to further clarify the key processing modules for improving accuracy. The results show that data augmentation and post-processing significantly improve the score in our system. In particular, mix-up data augmentation and beam search in post-processing improve SPIDEr by 0.8 and 1.6 points, respectively.

Index Terms— Automated audio captioning, data augmentation, multi-task learning, beam search decoding

1. INTRODUCTION

Automated audio captioning (AAC) is a crossmodal translation task in which input audio is translated into a description of the audio using natural language [1–6]. Whereas automatic speech recognition (ASR) converts speech to text, AAC converts environmental sounds to text. Generating meaningful captions for environmental sounds requires the incorporation of higher contextual information, including concepts, physical properties, and high-level knowledge, though such information is not necessarily needed for tagging scenes [7, 8], events [9], and conditions [10].

For the AAC task of the Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge 2020, Drossos et al. set up Task 6 [11], which prohibited the use of external resources and pre-trained models. Thus, participants were required to solve task-specific problems for audio captioning, rather than provide computational generic solutions using large amounts of training data or large-scale pre-trained models such as VGGish [12] and BERT [13]. This is because, even though image or video captioning is being actively studied, AAC is still an emerging research field, and because there has not been an exhaustive study on methods for improving accuracy.

Here, we address three research questions in AAC. The first is how to augment the limited data for effective training. Therefore, we discuss data augmentation that makes effective use of small amounts of data. The second is how to decide the best description for given audio. As described in [5, 6], information contained in audio signals can be much more ambiguous than that in images or videos. Moreover, the validity of the description generally depends on the situation or context as well as the sound itself. Therefore, we

Table 1: The three elements and their component modules used in our system [14]

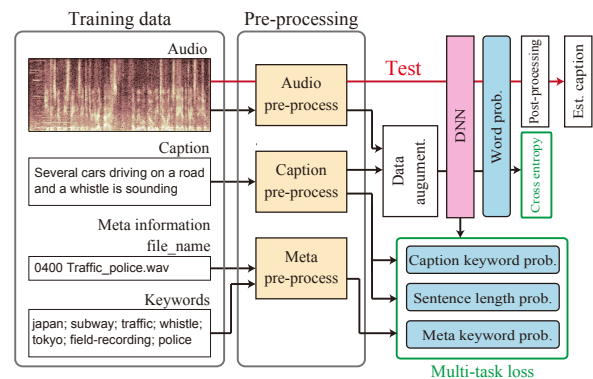|	
(i) Data augmentation	(a) Mix-up [15]
	(b) TF-IDF-based word placement [16]
	(c) IDF-based sample selection
(ii) Multi-task learning	(a) Sentence length estimation
	(b) Keyword estimation [6]
(iii) Post-processing	(a) Beam search decoding [17, 18]
	(b) Test time augmentation [19]

Figure 1: System overview.

introduce a deep neural network (DNN) architecture for estimating keywords and sentence length along with caption generation as multi-task learning. The third is whether post-processing such as beam search decoding is effective in AAC. This is because it has been found that post-processing can correct obviously incorrect output in machine translation and image captioning.

Figure 1 shows an overview of our system which incorporates data augmentation, a DNN architecture with multi-task learning, and post-processing, as summarized in Table 1. Although the system yielded the best score at the challenge, the effectiveness of each module has not yet been clarified.

This paper is organized as follows. In Section 2, we describe in detail the research questions that were raised through this challenge and explain our motivation for adopting each module. As described in Section 3, we carried out an ablation study. First, we roughly divide our system into three processing elements for ablation, (i) data augmentation, (ii) multi-task learning, and (iii) post-processing. Based on the result showing that (i) and (iii) mainly contribute to the accuracy, we then conducted a finer ablation study for each component module of (i) and (iii). Our findings are summarized as follows:

1. Data augmentation and post-processing significantly im-

proved the performance of AAC.

2. Multi-task learning did not improve the performance under the task rules, even though it was effective in more computationally rich scenarios, i.e., with pre-trained models [6].

3. Mix-up data augmentation was effective as with acoustic event detection and acoustic scene classification.

4. Beam search decoding was effective for AAC as well as other text generation tasks including ASR [20] and image captioning [21].

2. SYSTEM DESCRIPTION

This section describes our research questions and their solutions. See our technical report for the implementation details.

2.1. Research questions in AAC

We identified three research questions in AAC:

Data augmentation: In AAC, unlike other language processing tasks such as machine translation, it is difficult to collect training data from the Web. Therefore, the amount of training data available is inevitably less than that for other natural language processing tasks. In fact, there are only 14,465 training captions in the Clotho dataset [2], whereas there are 36M in the WMT 2014 English-French dataset for machine translation. This leads to the first research question: *how to augment the limited data for effective training*.

Multi-task training for resolving indeterminacy: As described in our previous studies, AAC has two indeterminacy problems. One is the indeterminacy in word selection [6], and the other is that in the degree of detail and sentence length [5]. The first problem arises because one acoustic event/scene can be described in several different ways using different sets of words, such as {*car, automobile, vehicle, wheels*} and {*road, roadway, intersection, street*} [6]. The second problem appears because a sound can be explained briefly or in detail with a long expression, such as "car sounds," or "small cars and large trucks are driving on a roadway and they are making very loud engine noises" [5]. Such indeterminacy leads to a combinatorial explosion of possible answers, making it almost impossible to estimate the best description and appropriately train AAC systems.

Post-processing: Post-processing such as beam search decoding is effective for improving accuracy. Post-processing in natural language processing can correct obviously incorrect output based on some explicit criteria. In DNN-based text generation, some post-processing methods also modify the output of the DNN to improve accuracy. Since AAC is a crossmodal text generation task with audio input, it is worth investigating whether post-processing is also effective in AAC and what kind of post-processing is effective.

2.2. Data augmentation

IDF-based sample selection: Since the objective metrics for captioning such as CIDEr [22] and SPIDEr [23] focus on the accuracy of infrequent words and idioms, we adopt two tricks for selecting the training samples based on IDF.

The first one is used for selecting an audio sample x from the training dataset. First, we concatenate five ground-truth captions corresponding to each audio clip x in the training dataset and use the concatenated captions as a "sentence". Then, we calculate the IDFs for all words in all sentences and the average IDF of each sentence. Finally, each average IDF is normalized by the sum of the
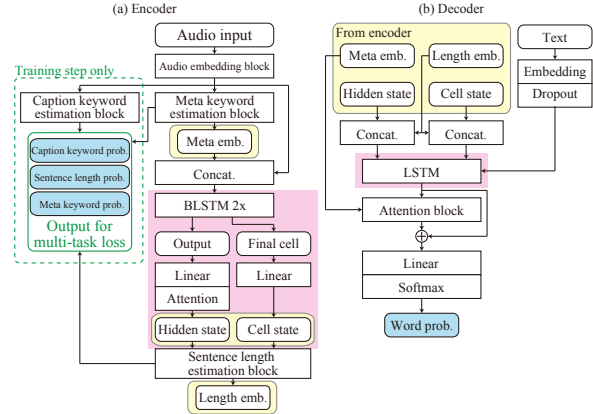

Figure 2: Architecture of captioning DNN.

average IDFs. We regard the normalized IDF as the parameter of the categorical distribution and select x based on this probability. Since data with a high average IDF contain low frequency words in the dataset, it is assumed that the data contain low frequency topics. Increasing the probability of selecting low frequency topics is expected to prevent over-fitting on high frequency topics.

The second trick is used for selecting a ground-truth caption w from five captions corresponding to the selected x. The basic strategy is the same as in the first trick. First, we calculate the IDFs of all words in the five captions. The document used here comprises the five captions, while the document used in the first trick consists of all sentences of the dataset. Then, we calculate the normalized IDF and use that as the parameter of categorical distribution. The target caption w is selected based on this probability. By increasing the probability that low frequency words are selected, we expect to avoid over-fitting on high frequency words.

TF-IDF-based word placement: Considering the limited number of sentences that can be used for training AAC systems, sentences in the dataset should be augmented in some way. One simple data augmentation method is to replace the words randomly. However, the meaning of sentences can change dramatically if informative words are replaced, and the correspondence between the caption and the audio clip will collapse. To cope with this problem, we need to control the words to be replaced. Thus, we adopt the TF-IDF based word placement method [16] to augment texts. This method avoids replacing informative words by limiting the words to be replaced to those with low TF-IDF and augments text data without largely changing the meaning of sentences.

Mix-up: For audio augmentation, we also investigate whether the mix-up that has been validated in past DCASE challenge tasks is effective. Since text is input as a set of class labels, direct mixing of w is not suitable. Thus, the embedded word tokens are mixed by multiplying the mixing parameters [24].

2.3. Multi-task training for resolving indeterminacy

To solve the indeterminacy problems, we build a DNN architecture and loss functions based on multi-task learning to simultaneously estimate the caption keywords, meta keywords and sentence length. Figure 2 shows the architecture of the captioning DNN. The pink area in Fig. 2 is a basic sequence-to-sequence (Seq2Seq)–based captioning model [25] using bidirectional long short-term memory (BLSTM)–LSTM. The output and final cell state of the encoder BLSTM are used for the initial hidden state and cell state of the

decoder LSTM. Then, the decoder LSTM estimates the posterior probability of the nth word given the audio signal $\boldsymbol{x}$ and the 1st to $(n-1)$-th words $p(w_n|\boldsymbol{x}, w_{1,...,n-1})$ using embedded word tokens. In addition, we use sub-blocks for keyword and sentence length estimation to solve the indeterminacy problems in AAC. The following describes an overview of these sub-blocks. Please refer to our technical report [14] for the details of their implementations.

Meta keyword estimation: The meta keyword estimation block also follows the audio embedding block, and its output is passed to the encoder. We expect this block to estimate keywords that humans associate with sounds, such as information from vision and the surrounding environment and thus avoid the indeterminacy of word selection.

In this study, the meta keywords were extracted from the `file_name` and `keyword` provided in the metadata CSV file, because they indicate their content, and a short textual description. We extracted the keywords using a keyword vocabulary that was manually created beforehand. The procedure for creating the keyword vocabulary is as follows. First, `file_name` and `keyword` were split at spaces and punctuation marks. Next, words that seem to be nouns, verbs, adjectives, and adverbs were converted to their lemmas. Finally, all those were counted, and lemmas that appeared more than ten times were appended in the keyword vocabulary, which is a hash table that maps the original word to its lemma.

The meta keyword estimation block is used to avoid the indeterminacy of word selection in the same way that the caption keyword estimation block is used. This block estimates meta keyword probabilities from the output of the audio embedding block. The weight of these losses are based on the prior probability of the ith keyword calculated by

$$\frac{\text{\# of }i\text{th keyword in training samples}}{\text{\# of training samples}}. \quad (1)$$

By using the inverse of this prior probability as the weight, the word and topic frequencies are considered in the training phase.

Caption keyword estimation: The caption keyword estimation block follows the audio embedding block. In contrast to the meta keyword estimation block, this block is not used in the testing phase. We expect this block to guide the training of the audio embedding block so that its output includes information about the keywords of the ground-truth caption. This also results in avoiding the indeterminacy of word selection.

This block estimates the caption keyword probabilities from the output of the audio embedding block. The caption keywords are the lemma of words frequently used in the provided meta data. The loss function for this block is the weighted binary-cross entropy. The weight of the loss is Eq. (1).

Sentence length estimation: The sentence length estimation is used to avoid the indeterminacy of sentence length. This block estimates sentence length probability and its embedding from the output and final cell state of the encoder BLSTM. The sentence length embedding is passed to the decoder to generate the sentence using the length information as a part of the initial hidden state and the cell state of the decoder LSTM.

We use the softmax cross entropy as the loss for the sentence length estimation block. By using these losses for multi-task learning, we expect the common audio embedding block to be learned as a more general representation for any task and that the AAC scores are improved.

Keyword co-occurrence loss: To prevent the decoder from outputting words obviously unrelated to the meta keywords, we additionally use the keyword co-occurrence loss between words in a caption and its meta keywords. For example, when meta keywords are {*car, sing, bird*}, words not related to keywords such as {*people, children, talking, talk, speak*} may not be included in the correct caption. To prevent the decoder from outputting such words, we adopt a penalty based on the decoder outputs $p(w_n|\boldsymbol{x}, w_{1,...,n-1})$. To implement this loss, we create a hash-table of the co-occurrence lists before training. Then, in the training phase, we add the penalties of the decoder outputs to the whole loss value.

2.4. Post-processing

We use beam search decoding for making word decisions process from $p(w_n|\boldsymbol{x}, w_{1,...,n-1})$. The beam size is 5, and n-gram blocking size is 2, i.e., a hypothesis in a beam is discarded if a bi-gram appears more than once within it. In addition, we use test time augmentation (TTA) for audio input. This is because the audio input is randomly cropped to limit the length of time in the training phase. If changing the length of the audio input in the testing phase could have a bad influence on the batch normalization layers. Therefore, in the testing phase, we also randomly crop and zero-pad the audio input so that all the inputs have the same length. We generate five input audio clips by this process, and take the average of the five outputs of the decoder.

3. EXPERIMENTS

3.1. Experimental setup

As the training and test dataset, we used the Clotho dataset [2], which consists of audio clips from the Freesound platform [26] and whose captions were annotated via crowdsourcing [27]. We used the development split of 2,893 audio clips with 14,465 captions (i.e., one audio clip has five ground-truth captions) for training and the evaluation split of 1,045 audio clips with 5,225 captions for testing. From the development split, 96 audio clips and their captions were randomly selected as the validation split. The setups of the DNN architecture and training were the same as in [14]

To evaluate our system, we use the same metrics as Task 6: BLEU-1, BLEU-2, BLEU-3, BLEU-4, ROUGE-L, METEOR, CIDEr [22], SPICE [28], and SPIDEr [23]. Among these scores, the scores developed for captioning task are CIDEr, SPICE, and SPIDEr. CIDEr evaluates TF-IDF weighted n-gram similarity between the output sentence and the reference sentences, which ensures generated captions are syntactically fluent, and SPICE compares grammatically parsed generated and references sentences, which ensures generated captions are semantically faithful to the audio. SPIDEr is the linear combination of CIDEr and SPICE, and it was used as the target metric in this challenge.

3.2. Experiment-I: Element-wise ablation study

We first conducted an element-wise ablation study; the three processing elements (i) data augmentation, (ii) multi-task learning, and (iii) post-processing were excluded one each, and the accuracy with the remaining two was compared. In addition, we also evaluated the performance when all the three elements were excluded, namely, the performance of a pure seq2seq model. We trained each model three times with different initial values, and calculated the average scores. We expected this experiment to clarify which processing elements have a significant impact on accuracy and help us find the processing element to focus on in future AAC studies.

Table 2: Results of element-wise ablation study

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr
Baseline of DCASE2020 Task6 [11]	38.9	13.6	5.5	1.5	26.2	8.4	7.4	3.3	5.4
Proposed	**51.2**	32.5	21.5	14.0	34.3	14.5	29.0	8.9	19.0
w/o (i) Data augmentaion	52.0	31.2	20.0	12.7	**33.7**	14.0	26.1	**8.2**	**17.2**
w/o (ii) Multi-task learning	51.8	33.0	21.7	14.1	34.6	14.7	29.1	9.2	19.2
w/o (iii) Post-processing	51.8	**30.1**	**18.0**	**10.8**	33.8	**13.9**	**25.3**	9.0	**17.2**
w/o all three elements	52.1	29.4	17.4	10.3	33.5	13.8	23.2	8.5	15.8

Table 3: Results of module-wise ablation study

Model	B-1	B-2	B-3	B-4	METEOR	ROUGE-L	CIDEr	SPICE	SPIDEr
(i) Data augmentation									
w/o (a) Mix-up	51.5	31.5	20.4	13.1	**33.6**	14.3	27.9	**8.4**	18.2
w/o (b) TF-IDF-based word placement	51.9	32.6	21.4	13.9	34.5	14.6	28.9	8.8	18.8
w/o (c) IDF-based sample selection	**51.1**	32.4	21.6	14.2	34.2	14.5	29.3	8.9	19.1
(iii) Post-processing									
w/o (a) Beam search decoding	52.1	**30.3**	**18.1**	**10.8**	33.9	**14.0**	**25.6**	9.2	**17.4**
w/o (b)TTA	51.5	32.8	21.8	14.3	34.2	14.5	29.4	8.9	19.2

Table 2 shows the results of the element-wise ablation study. `Baseline` means the scores of the baseline model for the DCASE 2020 Challenge Task 6 [1], and `Proposed` is the full model used in our submission. These results suggest the following:

Effective elements: (i) Data augmentation and (iii) post-processing elements significantly contributed to the performance of our submission. When one of these elements was removed, the SPIDEr score decreased by 1.8 points. In addition, the SPIDEr score decreased by 3.2 points when all the elements were removed. Thus, it can be considered that the effects of these elements are independent of each other and that the combination of these elements is effective.

Non-effective element: (ii) Multi-task learning element was not very effective under this primitive rule. Surprisingly, even though this element has been proven to be effective in more computationally intensive scenarios, such as ones with pre-trained models [6], it became clear that it was not effective in our submission. Since we simultaneously trained all the DNN modules from scratch using multi-task losses, they were probably not trained effectively. For these modules to be effective, pre-training might be required, especially when the number of the trainable parameters is large, as it was in our submission.

3.3. Experiment-II: Module-wise ablation study

Next, we conducted a module-wise ablation study to assess the contribution of each module.We calculated the scores of the models that any of the modules in processing blocks (i) and (iii) were excluded one each, where (i) data augmentation and (iii) post-processing blocks include modules {mix-up, TF-IDF-based word placement, IDF-based sample selection} and {beam search decoding, TTA}, respectively. As in Experiment-I, we trained each model three times with different initial values, and calculated the average scores.

Table 3 shows the results of the module-wise ablation study, and which written in the same manner of the Experiment-I. These results suggest us the following three things:

(I) The mix-up was effective, as used in acoustic event detection and acoustic scene classification.

(II) The beam search decoding worked well, as used in other

text generation tasks. Especially the CIDEr score, which depends on TF-IDF, increased. Since the AAC system without beam search decoding maximizes the posterior probability of each word, most of the output word sequences is high-frequency words in the dataset, not an optimal sentence sequence. Therefore, it can be considered that the beam search decoding has little impact on uni-gram-based metrics such as BLEU-1 and SPICE and improves the metrics that evaluate from the perspective of the whole sentence such as BLEU-4 and CIDEr.

(III) The use of the IDF-based sample selection and TTA did not improve or worsen the performance. The reason for this may be that the parameters we used in these methods in our implementation had little effect on the statistics of the augmented data. In TTA, the length of cropped samples was 20 sec, although that of training/testing samples was also around 20 sec. Thus, all the multiple cropped inputs have almost the same statistics as the original one, and in which would explain why TTA did not work effectively. In addition, in the Clotho dataset, to keep reference captions consistent, the words that appear in captions are intentionally designed lowly diverse. Therefore, simple sample or word replacement would not have been sufficient to augment training samples. To drastically vary the statistics of the augmented data, we may need to use para-phrase generation [29, 30] or thesaurus-based word placement [31].

4. CONCLUSIONS

In this paper, we presented an audio captioning system with data augmentation, multi-task learning, and post processing, and conducted a detailed ablation study to clarify which elements are effective. From the results, mix-up data augmentation and beam search decoding were effective in improving the accuracy of AAC. In particular, since beam search decoding selects output from the perspective of the whole sentence, BLEU-4 and CIDEr, which reflect n-gram accuracy, were significantly improved. Future works include the use of data augmentation that maintains grammatical structure such as para-phrase generation for sufficient augmentation of training samples.

5. REFERENCES

[1] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks," in *IEEE Workshop Appl. Signal Process, Audio Acoust, (WASPAA)*, 2017, pp. 374–378.

[2] ——, "Clotho: An audio captioning dataset," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 736–740.

[3] M. Wu, H. Dinkel, and K. Yu, "Audio caption: Listen and tell," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2019, pp. 830–834.

[4] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *N. Am. Chapter Assoc. Comput. Linguist.: Hum. Lang, Tech, (NAACL-HLT)*, 2019, pp. 119–132.

[5] S. Ikawa and K. Kashino, "Neural audio captioning based on conditional sequence-to-sequence model," in *Detect. Classif. Acoust, Scenes Events Workshop (DCASE)*, 2019, pp. 99–103.

[6] Y. Koizumi, R. Masumura, K. Nishida, M. Yasuda, , and S. Saito, "A transformer-based audio captioning model with keyword estimation," submitted to *Interspeech*, 2020.

[7] A. Mesaros, T. Heittola, A. Eronen, and T. Virtanen, "Acoustic event detection in real life recordings," in *Eur. Signal Process. Conf. (EUSIPCO)*, 2010, pp. 1267–1271.

[8] K. Imoto, N. Tonami, Y. Koizumi, M. Yasuda, R. Yamanishi, and Y. Yamashita, "Sound event detection by multitask learning of sound events and scenes with soft scene labels," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2020, pp. 621–625.

[9] D. Barchiesi, D. Giannoulis, D. Stowell, and M. D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 16–34, 2015.

[10] Y. Koizumi, S. Saito, H. Uematsu, Y. Kawachi, and N. Harada, "Unsupervised detection of anomalous sound based on deep learning and the neyman–pearson lemma," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 1, pp. 212–224, 2018.

[11] DCASE2020 Challenge Task 6: Automated Audio Captioning, http://dcase.community/challenge2020/task-automatic-audio-captioning.

[12] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss, and K. Wilson, "CNN architectures for largescale audio classification," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 131–135.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] 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., 2020.

[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Int. Conf. Learn. Represent.*, 2018.

[16] Q. Xie, Z. Dai, E. Hovy, M.-T. Luong, and Q. V. Le, "A transformer-based audio captioning model with keyword estimation," *arXiv preprint, arXiv:1904.12848*, 2019.

[17] P. Koehn, *Statistical machine translation.* Cambridge University Press, 2009.

[18] ——, "Pharaoh: a beam search decoder for phrase-based statistical machine translation," in *Mach. Transl.: Real Users Res. Proc. 6th Conf. Assoc. Mach. Transl. Am. (AMTA-2004)*, vol. 3265, 2004.

[19] M. S. Ayhan and P. Berens, "Test-time data augmentation for estimation of heteroscedastic aleatoric uncertainty in deep neural networks," in *Med. Imaging Deep Learn. (MIDL)*, 2018.

[20] H. Seki, T. Hori, S. Watanabe, N. Moritz, and J. Le Roux, "Vectorized beam search for ctc-attention-based speech recognition." in *INTERSPEECH*, 2019, pp. 3825–3829.

[21] B. Anderson, P.and Fernando, M. Johnson, and S. Gould, "Guided open vocabulary image captioning with constrained beam search," in *Conf. Empir. Methods Nat. Lang. Process.*, 2017, pp. 936–945.

[22] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *IEEE conf. comput. vis. pattern recognit. (CVPR)*, 2015, pp. 4566–4575.

[23] S. Liu, Z. Zhu, N. Ye, S. Guadarrama, and K. Murphy, "Improved image captioning via policy gradient optimization of spider," in *IEEE int. conf. comput. vis.*, 2017, pp. 873–881.

[24] H. Guo, Y. Mao, and R. Zhang, "Augmenting data with mixup for sentence classification: An empirical study," *arXiv preprint arXiv:1905.08941*, 2019.

[25] I. Sutskever, O. Vinyals, , and Q. V. Le, "Sequence to sequence learning with neural networks," in *Adv. Neural Inf. Process, Syst. (NIPS)*, 2014, pp. 3104–3112.

[26] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Int. Conf. Multimedia (MM'13)*, 2013, pp. 411–412.

[27] S. Lipping, K. Drossos, and T. Virtanen, "Crowdsourcing a dataset of audio captions," in *Detect. Classif. Acoust, Scenes Events Workshop (DCASE)*, 2019, pp. 139–143.

[28] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "SPICE: Semantic propositional image caption evaluation," in *Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 382–398.

[29] A. Sokolov and D. Filimonov, "Neural machine translation for paraphrase generation," *arXiv preprint arXiv:2006.14223*, 2020.

[30] M. Lewis, M. Ghazvininejad, G. Ghosh, A. Aghajanyan, S. Wang, and L. Zettlemoyer, "Pre-training via paraphrasing," *arXiv preprint arXiv:2006.15020*, 2020.

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