AUDIO CAPTIONING BASED ON TRANSFORMER AND PRE-TRAINED CNN

Kun Chen[1], *Yusong Wu*[1], *Ziyue Wang*[2], *Xuan Zhang*[2], *Fudong Nian*[3], *Shengchen Li*[1], *Xi Shao*[2]

[1] Beijing University of Posts and Telecommunications, Beijing, China,
{cksy1118, wuyusong, shengchen.li}@bupt.edu.cn
[2] Nanjing University of Posts and Telecommunications, Nanjing, China,
{1218012222, 1218012223, shaoxi}@njupt.edu.cn
[3] Anhui University, Anhui, China,
nianfudong@ahu.edu.cn

ABSTRACT

Automated audio captioning is the task that generates text description of a piece of audio. This paper proposes a solution of automated audio captioning based on a combination of pre-trained CNN layers and a sequence-to-sequence architecture based on Transformer. The pre-trained CNN layers are adopted from a CNN based neural network for acoustic event tagging, which makes the latent variable resulted more efficient on generating captions. Transformer decoder is used in the sequence-to-sequence architecture as a consequence of comparing the performance of the more classical LSTM layers. The proposed system achieves a SPIDEr score of 0.227 for the DCASE challenge 2020 Task 6 with data augmentation and label smoothing applied.

***Index Terms*—** audio captioning, acoustic event detection, transformer, data augmentation, label smoothing

1. INTRODUCTION

Automated audio captioning is a multimodal translation task where the model outputs a textual description given an audio signal. With growing attention [1–6], the 2020 edition of DCASE (Detection and Classification of Acoustic Scenes and Events) data challenge has introduced a task of automated audio captioning.

The basic tasks of automated audio captioning is acoustic event tagging and acoustic scenes identification. Besides, the automated audio captioning systems should also establish mappings between natural language and detected acoustic events. Interested mappings include 1) spatiotemporal relationships of sources (e.g. turned on the gas on a gas canister and **then** switched the ignition key), 2) foreground versus background discrimination (e.g. machine running and people talking in the **back**), 3) concepts (e.g. **muffled** sound) and 4) physical properties of objects and environment (e.g. **hard** surface, **wooden** door) [4].

Similar to image captioning [7–11], audio captioning requires to extract feature representations of input space and map into natural language space. For audio captioning, effective audio feature extraction methods [2] can extract accurate audio features, which makes the generated description more relevant to the audio. As the data available in the audio captioning task is limited, direct training from scratch in an end-to-end manner may not enough to train an effective feature extractor. Thus, A method of pre-training feature extraction network is introduced in audio captioning models. Meanwhile, it is essential to describe the audio information with accurate and fluent text in audio captioning task, which needs to establish a effective mapping between language and acoustic features. In natural language processing, Transformer [12] is a excellent language modeling architecture proposed in recent years. For better mapping the relationship between language and audio features, Transformer is introduced to replace the traditional decoder with LSTM layers in sequence-to-sequence architecture.

In DCASE 2020 audio captioning challenge, external data and pre-trained models are not allowed. Effectively training a feature extractor only using a small amount of data and annotation provided by the dataset is important. In this paper, a sequence-to-sequence model[1] is proposed to be trained using only caption annotation. The proposed model consists of a 10-layer CNN [13] encoder and a Transformer [12] decoder for feature extraction and natural language generation. To achieve better feature extraction and language modeling, pre-training is applied on the CNN encoder using the data provided by the challenge. To further improve performance and prevent over-fitting, label smoothing and data augmentation are applied during training, while a fine-tuning with small learning rate is applied after training is finished.

The contributions of this work are in the following aspects. First, we propose a CNN-Transformer model for audio captioning and achieves better results than the CNN-LSTM model [7]. Second, we design some selection rules to get the right words from the captions and set up a multi-label classification task to pre-train the CNN encoder. Third, we find that Label smoothing [14] and SpecAugment [15] can improve our model performance and avoid over-fitting effectively. The paper is organized as follows: Section 2 introduces the related research in the field of audio captioning. Section 3 describes the proposed model for DCASE 2020 audio captioning challenge. Section 4 introduces the experiment setup. The experimental results are presented in Section 5. Section 6 concludes this work.

2. RELATED WORKS

One of the first approach to automated audio captioning was proposed by Drossos et al. [3], which used a GRU-based encoder-decoder architecture. In [1, 6], the task of audio captioning in specific scenes was explored. In DASE 2019, Ikawa et all. [5] introduced a conditional sequence-to-sequence model to control the information of the output sentences. In [2], a model with top-down multi-scale encoder and aligned-semantic attention was proposed to address the problem of audio captioning for sound in the wild, and

[1] `https://github.com/lukewys/dcase_2020_T6`.

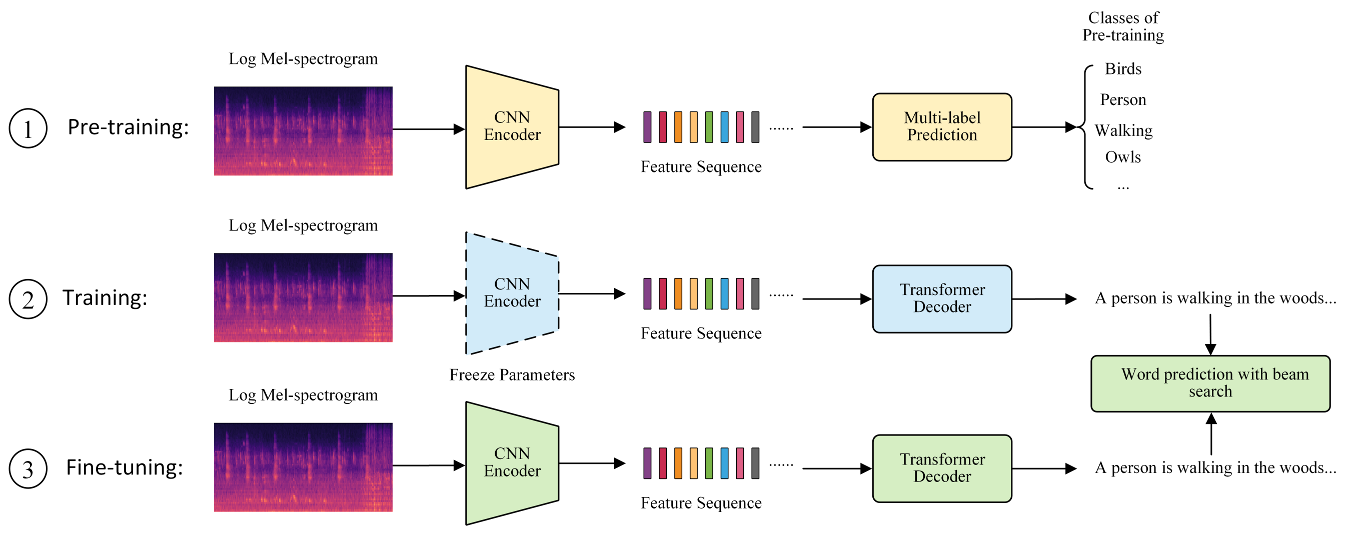

Figure 1: The overview diagram of the proposed model. The encoder extracts a sequence of feature vectors of the input log Mel-spectrogram, and the decoder generates each word while attending to the feature sequence. The encoder is firstly pre-trained by a multi-label prediction task. Fine-tuning is applied after training. The CNN encoder showed on the diagram remains the same architecture during pre-training, training and fine-tuning.

a large-scale dataset named AudioCaps was contributed. Recently, the dataset Clotho [4] was released and adopted in DCASE 2020 audio captioning challenge.

3. PROPOSED MODEL

The proposed model is a typical sequence-to-sequence system with the introduction of Transformer layers in the decoder part and more classical CNN layers in the encoder. The CNN layers, which is known as pre-trained CNN layers, in the encoder of the system are adopted in a separate audio tagging system, which mapping acoustic features with acoustic events extracted from the caption text in the training dataset.

3.1. Encoder

A 10-layer CNN [13] (CNN10) is used as the encoder in the proposed model. The CNN used here is for feature extraction of input spectrogram. As demonstrated to be successful in audio pattern recognition [13], CNN10 in [13] is adapted and used. The reason for choosing a relatively simple CNN rather than a deeper CNN such as VGG [16] or ResNet [17] is mainly to prevent over-fitting.

The 10-layer CNN consists of four convolution blocks each having two 3×3 convolution layers with ReLU activation function and batch normalization, with 2×2 average pooling layer between the blocks. The number of channels in the convolution blocks is 64, 128, 256, 512, respectively.

The output of the CNN is a 512 channel feature sequence downsampled 16 times both in the time dimension and frequency dimension. Each feature vector in the frequency dimension is then averaged. The 2 layers of fully-connected neural networks are used to map the dimension of each vector in feature sequence into the number of hidden dimensions used in the decoder for attention computation.

3.2. Decoder

The decoder used in the proposed model is a standard Transformer [12] consist of multi-head self-attention on text sequence and multi-head encoder-decoder attention on extracted feature sequence. The reason for choosing the Transformer model as the decoder is because of its state-of-the-art performance on natural language processing and the non-recurrence computing of its structure which helps prevent gradient vanishing or exploding. The decoder uses a 2-layer Transformer with a hidden dimension of 192 and 4 heads.

3.3. Word prediction

Cross-entropy loss is used for this Audio captioning model. For the word w_n of the target caption, the corresponding output is $Dec(w_{1,2,\ldots,n-1}, x) \in R^N$, where x represents the output of the encoder, Dec stands for decoder, N is the vocabulary size. For prediction, the $softmax$ function is used to turn the output into word probability. In the inference stage, the current prediction is related to the prediction results of the previous steps. Beam search with a size of 1-4 is used for word inference and chooses the best beam size based on the output caption's score. When the beam size is 1, the model selects the word with the highest probability as the output at each step.

4. EXPERIMENTS

The data used in this paper is DCASE 2020 with essential preprocess steps. Then the proposed model is trained by a three-stage strategy as shown in Fig. 1.

4.1. Data pre-processing

In DCASE 2020 audio captioning challenge, Clotho dataset [4] is used which contains 4981 audio clips in 15-30 seconds each with 5

Model	BLEU_1	BLEU_2	BLEU_3	BLEU_4	ROUGE_L	METEOR	CIDEr	SPICE	**SPIDEr**
Baseline	0.389	0.136	0.055	0.015	0.262	0.084	0.074	0.033	0.054
PreCNN-LSTM	0.487	0.294	0.184	0.114	0.340	0.158	0.272	0.102	0.187
CNN-Transformer	0.528	0.324	0.211	0.135	0.346	0.150	0.300	0.099	0.199
PreCNN-Transformer	0.524	0.326	0.201	0.134	0.350	0.154	0.314	0.105	**0.209**
Fine-tune PreCNN-Transformer	0.534	0.343	0.230	0.151	0.356	0.160	0.346	0.108	**0.227**

Table 1: Scores of each metric with different model on evaluation data. The Baseline model is the baseline of the audio captioning challenge. The PreCNN-Transformer model and the PreCNN-LSTM model use the same 10-layer CNN encoder [13] and compared with the same CNN pre-training weights.

captions in 8-20 English words. The text information and acoustic information are processed separately.

Experiments use log Mel-spectrograms as acoustic feature, which calculated from the raw audio. The log Mel-spectrogram input is obtained by first getting the 64 Mel-band Mel-spectrogram of the audio, then converting the amplitude into a decibel scale. The sampling rate of audio is 44100, the length of FFT window is 1024, and the number of samples between successive frames is 512. When input into the network, zero-padded the data in a batch and form a tensor.

Captions are pre-processed in our experiments. All captions were tokenized while remove all punctuations and convert all letters to lowcase. Then, add tokens at the start and the end of each caption to mark the sentence. Eventually, the vocabulary size of captions is 4368.

SpecAugment [15] is applied as data augmentation to increase the effective size of existing training data and demonstrated to be effective in performance improvement in automatic speech recognition models [15]. In SpecAugment, frequency masks and time masks are randomly applied onto the log Mel- spectrogram before input to the CNN encoder.SpecAugment is set with 2 frequency masks and 2 time masks in parameter $W = 40, T = 30$ with a probability of 0.2.

4.2. Pre-training

The CNN encoder in the model plays an important role in extracting features of the input audio. However, direct training may not be sufficient to train the encoder which makes decoder hard to optimize. Thus, to more effectively optimize the decoder during training, pre-training is applied here before training. The CNN encoder is pre-trained by converting the audio captioning task into a multi-label classification task where 300 classes are used. The classes used in encoder pre-training is obtained by selecting a sub-set of the word in caption vocabulary. First, 20 words with the highest frequency and the words that have no more than 2 letters are excluded from the vocabulary which mainly contains meaningless words such as article words. words in caption vocabulary are then transformed into its original form by finding the stems of words such as "-ing", "-ly", "-d", "-s", etc, while the word frequency of the transformed words is added to the frequency of its original form. Last, 300 words with the highest frequency remain in vocabulary is selected as classes for pre-training. Some of the selected classes are as follows: "chirp", "someone", "person", "talk", "run", "walk", "sound", "noise", "object", etc.

In the pre-training stage, all 5 captions of each audio are combined to form one training label. Words in each caption are also transformed into its original form using the same rules above. The label of each audio is a multi-hot vector where each index of the word occurs in captions equal to 1.

The input of the multi-label classification network x is the log Mel-spectrogram of audio, and the output of the network $f(x) \in [0, 1]^K$ represents the the probability of K classes. The target $y \in \{0, 1\}^K$ is the label extracted from all 5 captions of audio. The loss of the network is calculated using binary cross-entropy. Since the multi-label classification network and the encoder of our audio captioning network using the same CNN layers, it is very simple to transfer the learned features.

The word embedding in the decoder is also pre-trained to improve language modeling performance. The word embedding is pre-trained using Word2Vec model [18] via python package `genism` [19]. Each caption sentence in the training set is used to form a training corpus.

The multi-label classification model is pre-trained for 60 epochs with a learning rate of 1×10^{-3}. The Word2Vec model is trained 1000 epochs with random parameter initialization.

4.3. Training

Among the data in the dataset, 60% are used as development (training) set and 20% are used as evaluation set while the last 20% remains as test set.

Batch size of 16 is used with a learning rate of 3×10^{-4} and a $l2$ regularization applied to all trainable parameters with factor $\lambda = 10^{-6}$. The encoder of the audio captioning model loads the weights trained on multi-label classification model and freezes these parameters. Label smoothing [14] is set with $\epsilon = 0.1$. The dropout probability $p = 0.2$ is applied to CNN encoder and Transformer decoder.

To improve performance and avoid over-fitting, label smoothing [14] is applied as regulation to improve the generalization of the model by introducing penalty to over-confident prediction. In label smoothing, the one-hot word prediction label is replaced by mixing of original distribution $q(k|x) = \delta_{k,y}$ with a uniform distribution $u(k) = 1/K$ where K is the number of word classes, such that

$$q'(k) = (1 - \epsilon)\delta_{k,y} + \frac{\epsilon}{K}. \quad (1)$$

4.4. Fine-tuning

In training phase, the experiment loads pre-trained weights and freezes CNN parameters to train the decoder. In order to optimize the encoder, fine-tuning is applied which was found to improve the

Methods	BLEU_1	BLEU_2	BLEU_3	BLEU_4	$\mathrm{ROUGE}_\mathrm{L}$	METEOR	CIDEr	SPICE	**SPIDEr**
Base	0.511	0.311	0.201	0.129	0.338	0.084	0.277	0.095	0.186
SA	0.498	0.305	0.199	0.128	0.332	0.146	0.286	0.096	0.191
LS	0.513	0.316	0.207	0.135	0.345	0.149	0.295	0.098	0.197
SA+LS	0.528	0.324	0.211	0.135	0.346	0.150	0.300	0.099	0.199
SA+LS+PW	0.507	0.311	0.204	0.132	0.342	0.153	0.295	0.099	0.197
SA+LS+PC	0.524	0.326	0.201	0.134	0.350	0.154	0.314	0.105	0.209
SA+LS+PC+FT	0.534	0.343	0.230	0.151	0.356	0.160	0.346	0.108	**0.227**

Table 2: Scores of each metric with different methods on evaluation data. *Base*: Train the network without using tricks. *SA*: Add SpecAugment. *LS*: Add Label smoothing. *PW*: Use pre-trained word embedding. *PC*: Use pre-trained CNN encoder. *FT*: Fine-tuning.

model performance. The model which yields the highest performance in evaluation is selected, and continue training for 20 epochs with a small learning rate of 10^{-4}.

5. RESULTS

5.1. Evaluation metrics

Caption evaluation is performed using the tools provided by the organizer of this challenge. Among the metrics used, BLEU_n [20] measures a modified n-gram precision. $\mathrm{ROUGE}_\mathrm{L}$ [21] measures a score based on the longest common subsequence. METEOR [22] measures a harmonic mean of weighted unigram precision and recall. CIDEr [23] measures a weighted cosine similarity of n-grams. SPICE [24] measures the F-score of semantic propositions extracted from caption and reference. SPIDEr [25] is the arithmetic mean between the SPICE score and the CIDEr score. The submitted results are ranked by the SPIDEr score. In all metrics used, higher scores indicate better performance. The Rank of the methods is performed according to the SPIDEr metric in DCASE 2020 audio captioning challenge.

5.2. Caption Generation

The performance of the proposed model is shown in Tab. 1. As shown by SPIDEr, the pre-trained CNN-Transformer structure outperforms the pre-trained CNN-LSTM structure as the transformer layers can better establish mappings between natural language and acoustic features. Moreover, the pre-trained CNN-Transformer structure outperforms the CNN-transformer without pre-train CNN layers by a SPIDEr score of 0.209, which demonstrate that the proposed pre-traing method can extract better acoustic features.

Experiments compared the performance of different model in Tab. 1. The Baseline model[2] is the baseline of the audio captioning challenge, which consists of a three-layer BiGRU encoder and a two-layer BiGRU decoder. The CNN-Transformer is the model proposed in this paper. The PreCNN-LSTM uses the same CNN encoder [13] as CNN-Transformer and a LSTM decoder [7] with attention. To compare the ability of different decoder and reduce the influence of CNN encoder, PreCNN-LSTM and PreCNN-Transformer are compared with the same pre-training CNN layers. The results show that the model proposed in this paper possesses much higher performance than Baseline. The Transformer decoder is better than LSTM decoder in all evaluation metrics, which means Transformer has better language modeling ability than LSTM in this task. Eventually, the proposed model can achieve a score of 0.227 with fine-tuning.

[2]The performance the baseline model is presented according to `http://dcase.community/challenge2020/task-automatic-audio-captioning`.

5.3. Ablation studies

In order to show the validity of the tricks and the methods in proposed model, ablation studies are used to verify the effectiveness of each step. All experiments were run under the same machine, environment and configuration. The results are shown in Tab. 2, most methods used in the studies are effective to improve model performance.

In Tab. 2. Method Base is the CNN-Transformer model without using tricks. Method SA adds SpecAugment [15] for data augmentation. Method LS uses Label smoothing [14] to prevent overfitting. Method PW adds pre-trained word embedding obtained through Word2Vec. Method PC uses pre-trained CNN encoder obtained through the multi-label classification task. Method FT is to fine-tune the model with a smaller learning rate.

The results demonstrate that using SpecAugment and Label smoothing can effectively improve performance and avoid overfitting. The word embedding trained on the corpus obtained from the training set does not help the model. The pre-trained encoder obtained through the multi-label classification task can be directly used in the audio captioning model and achieves a certain improvement. At the same time, pre-training can reduce the difficulty of model training and get better modeling ability. After training, finetuning is necessary to optimize the encoder to get the best result.

6. CONCLUSION

Automated Audio Captioning is a task that has not been studied adequately. This paper proposes a CNN-Transformer model with a pretraining stage introduced. Experimental results show that the features extracted from the multi-label classification task can be used directly for Automated audio captioning. The pre-training method improves the feature extraction and language modeling capability of the model and greatly improve the performance. By using different decoder, it demonstrates that Transformer has better language modeling ability than LSTM in this task. The experiments also verified the effectiveness of SpecAugment and Label smoothing introduced in the proposed model.

7. REFERENCES

[1] M. Wu, H. Dinkel, and K. Yu, “Audio caption: Listen and tell,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 830–834.

[2] C. D. Kim, B. Kim, H. Lee, and G. Kim, “Audiocaps: Generating captions for audios in the wild,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.

[3] 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)*. IEEE, 2017, pp. 374–378.

[4] K. Drossos, S. Lipping, and T. Virtanen, “Clotho: An audio captioning dataset,” in *45th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[5] S. Ikawa and K. Kashino, “Neural audio captioning based on conditional sequence-to-sequence model,” in *2019 DCASE Workshop*, 2019.

[6] X. Xu, H. Dinkel, M. Wu, and K. Yu, “What does a car-ssette tape tell?” *arXiv preprint arXiv:1905.13448*, 2019.

[7] 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.

[8] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, “Show, attend and tell: Neural image caption generation with visual attention,” in *International conference on machine learning*, 2015, pp. 2048–2057.

[9] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.

[10] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, “Image captioning with semantic attention,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4651–4659.

[11] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6077–6086.

[12] 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.

[13] 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,” *arXiv preprint arXiv:1912.10211*, 2019.

[14] 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.

[15] 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.

[16] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv preprint arXiv:1301.3781*, 2013.

[19] R. Řehůřek and P. Sojka, “Software Framework for Topic Modelling with Large Corpora,” in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[20] 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.

[21] C.-Y. Lin, “Rouge: a package for automatic evaluation of summaries,” in *Workshop on Text Summarization Branches Out, Post-Conference Workshop of ACL 2004, Barcelona, Spain*, July 2004. [Online]. Available: https://www.microsoft.com/en-us/research/publication/rouge-a-package-for-automatic-evaluation-of-summaries/

[22] S. Banerjee and A. Lavie, “Meteor: An automatic metric for mt evaluation with improved correlation with human judgments,” in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

[23] 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.

[24] 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.

[25] 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.

[26] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, “A comprehensive survey of deep learning for image captioning,” *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–36, 2019.

[27] 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