# AUTOMATED AUDIO CAPTIONING WITH WEAKLY SUPERVISED PRE-TRAINING AND WORD SELECTION METHODS

*Qichen Han*[*], *Weiqiang Yuan*[*], *Dong Liu, Xiang Li, Zhen Yang*

NetEase (Hangzhou) Network Co., Ltd., China,
{hanqichen, yuanweiqiang, hzliudong, hzlixiang, yangzhen1}@corp.netease.com

**ABSTRACT**

Audio captioning is a multi-modal task, focusing on generating a natural sentence to describe the content in an audio clip. This paper proposes a solution of automated audio captioning based on weakly supervised pre-training and word selection methods. Our solution focuses on solving two problems in automated audio captioning: data insufficiency and word selection indeterminacy. As the amount of training data is limited, we collect large-scale weakly labeled dataset from Web with heuristic methods. Then we pre-train the encoder-decoder models with this dataset followed by fine-tuning on the Clotho dataset. To solve the word selection indeterminacy problem, we use keywords extracted from captions of similar audios and audio tags produced by pre-trained audio tagging models to guide caption generation. The proposed system achieves the best SPIDEr score of 0.310 in the DCASE 2021 Challenge Task 6.

***Index Terms***— Audio captioning, encoder-decoder modeling, weakly supervised pre-training, audio similarity, audio tag

## 1. INTRODUCTION

The automated audio captioning (AAC) problem is defined as an intermodal translation task of automatically generating a textual description for an input audio signal [1]. This task needs information including identification of sound events, acoustic scenes, spatiotemporal relationships of sources, foreground versus background discrimination, concepts, and physical properties of objects and environment [2]. Audio captioning needs to extract the feature representation of audio space and map it to natural language space. Therefore, most of the previous works adopt encoder-decoder framework [3-6]. Our solution focuses on solving two problems in automated audio captioning: data insufficiency and word selection indeterminacy.

As the amount of training data in the audio captioning task is limited, training a well generalized end-to-end model is difficult. It is well-established that pre-training on large datasets followed by fine-tuning on target datasets boosts performance [7]. We use heuristic methods to collect a weakly labeled dataset for pretraining, which contains 65667 audios and corresponding captions. In addition, our system uses PANN's [8] architecture as an encoder, which is trained on the large-scale AudioSet [9] dataset.

In AAC task, one acoustic event/scene in an audio can be described with different words, leading to a combinatorial explosion of possible captions [3]. This word selection indeterminacy problem may lead to difficulty in training. We try two methods to tackle this problem. Firstly, considering that similar audios may have similar captions, we train a model to calculate the similarity between audios and use keywords extracted from the captions of similar audios to assist decoding. Secondly, we try to use audio tag information to assist decoding.

The contributions of this work are in the following aspects. Firstly, we propose a method to use pretrained PANN models as encoder and to pretrain the whole model on a large weakly labeled dataset. Secondly, to relieve the word selection indeterminacy, we introduce audio tags and caption keywords in the decoding stage. Thirdly, ablation studies are conducted to confirm the effectiveness of different strategies in the proposed approach.

The paper is organized as follows: Section 2 describes the proposed method for DCASE 2021 audio captioning challenge. Section 3 introduces the ablation study experiment setup. The experimental results are presented in Section 4. Section 5 concludes this work.

## 2. SYSTEM DESCRIPTION

This section describes our methods. Please refer to our technical report for more details [10].

### 2.1. Data augmentation

**Perturb audio data** In the Clotho dataset, each audio has five captions. Using audio augmentation methods such as speed perturbation [11] and reverberation [12], we perform a 5-fold augmentation of the Clotho dataset.

[*] These authors contributed equally to this work.

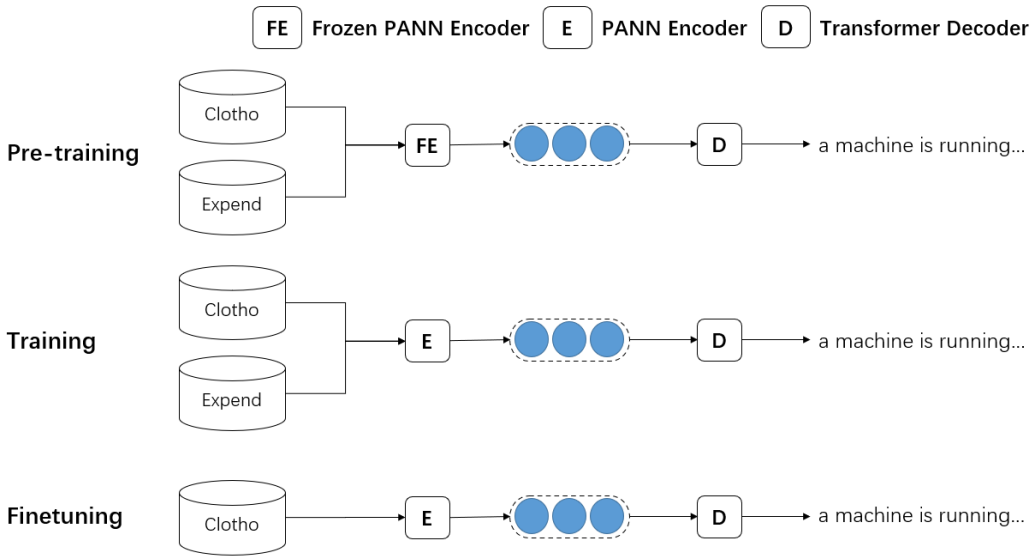


Figure 1: The overview diagram of the proposed method. The Expend data includes AudioCaps and weakly labeled dataset.

**AudioCaps dataset** We also add AudioCaps [13] for training, which is a large-scale dataset of about 51K audio clips to human-written text pairs collected via crowdsourcing on the AudioSet dataset.

**Weakly labeled dataset** We collect audios and corresponding descriptions from the Freesound[1], Zapsplat[2], Soundbible[3] and SoundJay[4] website. Audios that are shorter than 5 seconds are removed. And for those longer than 30 seconds, we randomly select 15 to 30 seconds of clips from the audio. We use heuristic rules to filter and clean captions [10]. As a result, we collected 65667 audios with captions from the four websites.

### 2.2. Pretrain encoder

PANNs are models pre-trained on raw AudioSet recordings with different structures. Several PANN systems outperform previous state-of-the-art audio tagging systems and can be transferred to other audio comprehension tasks. Three different networks in PANNs are selected as encoders, namely CNN14, Resnet38 and Wavegram-Logmel CNN[5].

### 2.3. Similar audio searching

As mentioned above, similar audios may have similar descriptions. For an audio without captions, we can get relevant keywords from the captions of its similar audios, which can help to generate better caption.

Inspired by text similarity calculation methods such as ESIM [14], we design a model to calculate the similarity between audios. We use CNN14[6] as audio encoder and get the 2048-dimension feature sequence. Then, we treat each audio feature sequence as the word embedding sequence and similarity between two audios is calculated with the ESIM network.

For training, we use SPIDEr score between captions of two audios as their ground truth similarity. We train this model with triplet dynamic margin loss. Given an anchor audio $a$, its similar audio $p$, and unsimilar audio $n$, the loss calculation is defined as follows:

$$Loss(a,p,n) = \max(0, m(a,p,n) + s(a,p) - s(a,n)) \quad (1)$$

where $s(.)$ is the similarity as mentioned above, and $m(.)$ is the margin function. The margins for each pair of (a, p, n) are different. We calculate the margins by the SPIDEr scores between captions of audios:

$$m(a,p,n) = \max(0.4, SPIDEr(a,p) - SPIDEr(a,n)) \quad (2)$$

### 2.4. Decoder

As in [4], we use transformer networks as decoder.

**Tag enhanced decoder** In order to reduce the search space, we utilize audio tag information by adding it to the beginning of the output sequence. The tags we use is based on AudioSet Ontology[7]. To avoid the problem of sparse data, we merge fine-grained tags with the help of the structural features of the ontology and get 13 tags, named Self-Tag-13. CNN14 is used to get the Self-Tag-13 tags of audios in training set and test set. During the training process, the decoder needs to predict the tag of audio before generating caption. In the test phase, the decoder generates caption given the corresponding tag of test audio.

**Keyword enhanced decoder** We extract keywords from captions of similar audios. Specifically, we select 50 captions of the top 10 most similar audios for the target audio, use NLTK[8] to perform stemming, and extract the top 10 keyword stems according to the TF-IDF weight. In the decoding stage, a fixed boost score is added to log likelihood for all word forms of the keywords. The boost score is set to 0.5.

[1] https://freesound.org
[2] https://www.zapsplat.com/
[3] https://soundbible.com/
[4] https://www.soundjay.com/
[5] https://zenodo.org/record/3987831#.YMhofqgzaUk
[6] https://github.com/qiuqiangkong/audioset_tagging_cnn
[7] http://research.google.com/audioset/ontology/index.html
[8] https://www.nltk.org/

Table 1: Ablation study. PE: Pre-trained encoder. KD: Keyword enhanced decoder. TD: Tag enhanced decoder. PD: Perturbed audio data. AD: AudioCaps dataset. WD: Weak label dataset.

| Model | $BLUE_1$ | $BLUE_2$ | $BLUE_3$ | $BLUE_4$ | METEOR | ROUGE-L | CIDEr | SPICE | SPIDEr |
|---|---|---|---|---|---|---|---|---|---|
| Baseline | 0.521 | 0.328 | 0.216 | 0.139 | 0.153 | 0.353 | 0.326 | 0.102 | 0.2142 |
| PE | 0.541 | 0.348 | 0.228 | 0.149 | 0.162 | 0.362 | 0.386 | 0.112 | 0.2490 |
| PE+KD | 0.552 | 0.360 | 0.240 | 0.156 | 0.167 | 0.372 | 0.409 | 0.119 | 0.2641 |
| PE+TD | 0.537 | 0.341 | 0.225 | 0.148 | 0.163 | 0.359 | 0.371 | 0.114 | 0.2427 |
| PE+PD | 0.550 | 0.353 | 0.232 | 0.149 | 0.164 | 0.366 | 0.385 | 0.118 | 0.2514 |
| PE+PD+AD | 0.554 | 0.356 | 0.235 | 0.153 | 0.167 | 0.364 | 0.405 | 0.117 | 0.2609 |
| PE+PD+AD+WD | 0.578 | 0.381 | 0.258 | 0.171 | 0.176 | 0.384 | 0.444 | 0.123 | 0.2837 |
| PE+PD+AD+WD+KD | **0.583** | **0.391** | **0.267** | **0.177** | **0.179** | **0.388** | **0.456** | **0.128** | **0.2920** |

Table 2: Experimental results on the evaluation split of Clotho dataset.

| Model | $BLUE_1$ | $BLUE_2$ | $BLUE_3$ | $BLUE_4$ | METEOR | ROUGE-L | CIDEr | SPICE | SPIDEr |
|---|---|---|---|---|---|---|---|---|---|
| Model 1: CNN14 | 0.583 | 0.388 | 0.265 | 0.178 | 0.179 | 0.385 | 0.473 | 0.128 | 0.300 |
| Model 2: Resnet38 | 0.593 | 0.400 | 0.274 | 0.184 | 0.183 | 0.392 | 0.482 | 0.133 | 0.308 |
| Model 3: Resnet38 + TD | 0.581 | 0.386 | 0.261 | 0.173 | 0.178 | 0.384 | 0.456 | 0.131 | 0.294 |
| Model 4: Wavegram-Logmel CNN | 0.585 | 0.392 | 0.269 | 0.182 | 0.177 | 0.389 | 0.474 | 0.130 | 0.302 |
| Ensemble 1 2 4 | 0.600 | 0.409 | 0.283 | 0.192 | 0.184 | 0.398 | 0.497 | 0.135 | 0.316 |
| Ensemble 1 2 3 4 | **0.603** | **0.414** | **0.286** | **0.195** | **0.186** | **0.400** | **0.499** | **0.137** | **0.318** |

## 3. EXPERIMENTS

### 3.1. Dataset

The Clotho [2] v2 dataset consists of audio clips from the Freesound platform [15] with captions annotated via crowdsourcing [16]. The Clotho v2 dataset is divided into a development split of 3839 audio clips, a validation split of 1045 audio clips, an evaluation split of 1045 audio clips, and a test split of 1043 audio clips.

We used the development split of Clotho, AudioCaps and weakly labeled dataset for training, the evaluation split for testing. The validation split is selected as the validation data.

### 3.2. Data pre-processing

All audio clips are down-sampled to 32kHz. The configuration of audio feature extraction is the same as that of PANNs[8]. We use words in the development-training split of Clotho as vocabulary. Words out of vocabulary are represented by <UNK>. <SOS> and <EOS> are also employed as the start-of-sequence and end-of-sequence tokens, respectively.

### 3.3. Training detail

**Training method** As is shown in Figure 1, the whole training process is divided into three stages. In the pre-training stage, the parameters of encoder are frozen and only the decoder is trained. In the training stage, the encoder parameters are unfrozen and trained together with the decoder. In the experiment where AudioCaps and weakly labeled dataset are included, the finetuning stage is used to finetune the model only with the Clotho dataset.

**Model settings** We use CNN14 as an encoder for ablation study, which consists of 6 convolutional blocks and each convolutional block consists of 2 convolutional layers with a kernel size of 3×3. In addition to CNN14, Resnet38 and Wavegram-Logmel CNN are used as encoder in our submissions. Resnet38 consists of 16 basic blocks in the Resnet [17], where each block consists of two convolutional layers with a kernel size of 3×3, and a shortcut connection between input and output. Wavegram-Logmel-CNN uses CNN14 as a backbone and uses a trainable 1D-Conv based frontend to extract features from time-domain waveforms. We use a 2-layer Transformer [18] with a hidden dimension of 256 and 4 heads as decoder.

To improve performance and avoid over-fitting, Label smoothing [19] and SpecAugment [20] are applied during training. The configuration of SpecAugment is the same as that of PANN. The learning rate is 3e-4, 1e-4 and 5e-5 for the three training stages separately. In the inference stage, a beam search with beam size 3 is implemented to achieve better decoding performance.

### 3.4. Evaluation metrics

A total of eight objective metrics are utilized to evaluate our model generated captions. Among the metrics used, BLEU@1-4 [21] measures a modified n-gram precision. METEOR [22] measures a harmonic mean of precision and recall of segments of the captions between the predicted and the target. ROUGEL [23] measures F-score based on the longest common subsequence. CIDEr [24] measures a weighted cosine similarity of n-grams. SPICE [25] compares semantic propositions extracted from caption and reference. SPIDEr [26] is the arithmetic mean between the SPICE score and the CIDEr score.

## 4. RESULTS

### 4.1. Ablation study

To verify the effectiveness of the tricks and components in the proposed model, several ablation experiments are conducted.

The experiment results are shown in Tab.1. Baseline is the model training from scratch with the Clotho dataset.

As shown in Table 1, most of the tricks and components can improve the final SPIDEr score. First of all, the benefits of data augmentation are significant. Both the AudioCaps dataset and the weakly labeled dataset collected from the Internet can improve the final performance. And the collected weakly labeled dataset can bring more benefit than the AudioCaps dataset. The reason may be that the weakly labeled dataset is collected from websites similar to Freesound, which matches the data distribution of Clotho dataset more closely.

Secondly, adding a pre-trained encoder can significantly improve SPIDEr scores, which shows that the pre-trained PANN can extract more effective features from the audio. Thirdly, perturbing audio data slightly improves the spider score.

Finally, the keyword enhanced decoder can assist the generation of captions. Compared to experiments without keyword enhanced decoder, both experiments with this component get great improvement on all evaluation metrics. This indicates that similar audio captions contain valuable information for AAC task.

Note that compared to using PE only, by adding the tag enhanced decoder, the SPIDEr score drops a little bit. We thought this decline was due to the tag prediction errors produced by pretrained PANN model.

### 4.2. Submitted systems

In Table 2, we present the relevant results of our submission 2, which is our best submission in the DCASE Challenge. To tackle the problem of insufficient data, validation data is added to the train dataset.

Three different model architectures of PANN are trained using the best strategy combination (PE+PD+AD+WD) obtained from ablation research. At the same time, we also try to add tag enhanced decoder to the Resnet architecture for training. Finally, model ensemble is performed by adding the predicted scores of multiple models. We first ensemble three different snapshots under the same model architecture and then we try two ensemble strategies, ensemble of three different architecture models and ensemble of all four architecture models.

Experiments show that Resnet achieves the highest spider score, and Resnet38 with tag enhanced decoder has the lowest score. Model ensemble can significantly improve SPIDEr scores. Although the SPIDEr score of the model with tag enhanced decoder declines, it improves the final result after the model ensemble.

### 4.3. Case study

We choose two cases from evaluation split of Clotho to show the impact of keyword enhanced decoder and tag enhanced decoder on captions generation. For each case, 2 representative reference captions are listed.

Table 3 shows the impact of keyword enhanced decoding. We can see that keyword enhanced decoding can make the caption more specific and closer to the caption written manually. The different forms of the ten keyword stems extracted from similar audio captions in the vocabulary is shown in the bottom of Table 3. Most of the keywords are in line with the content of the audio, which can help to generate captions more precisely.

Table 3: The case for Chopping pieces of mushrooms vigorously.wav.

| Name | Caption |
|---|---|
| Ref 1 | Vegetables are cut and chopped on a cutting board by someone. |
| Ref 2 | A person cutting and chopping vegetables on a cutting board. |
| w/o KD | chopping vegetables with a knife. |
| KD | chopping vegetables **on a cutting board** with a knife. |
| keyword | knives/knife chopping/chopped/chop/chops vegetable/vegetables woods/wood **cutting**/cuts/cut saw/saws/sawed/sawing/ boards/**board** wooden food slices/sliced/slicing |

Table 4: The case for SamyeLing_Pheasant121102.wav.

| Name | Caption |
|---|---|
| Ref 1 | A bird caws at regular intervals while smaller birds chirp in the background. |
| Ref 2 | A bird making a call and another bird that is chirping. |
| w/o TD | a person uses a tool to each other. |
| TD | a bird is chirping and then another bird is chirping in the background. |
| Tag | TGA_animal |

Table 4 shows an example of the advantage of the tag enhanced decoder. Guided by the tag, i.e., TGA_animal, captions about birds can be generated. Without this strategy, the generated captions are far from the reference captions. This case indicates that when the tag is accurate, the tag enhanced decoder can keep the generated caption within a reasonable space.

## 5. CONCLUSIONS

In this paper, we present a solution of automated audio captioning based on weakly supervised pre-training and word selection methods, and conducted a detailed ablation study to clarify which element is effective. From the results, pretrained encoder, keyword enhanced decoder and data augmentation are effective in improving the accuracy of AAC task. In particular, we propose a set of heuristic methods for collecting weakly-labeled data sets. This method can effectively alleviate the problem of insufficient data. We also verified that the captions of similar audio are valuable for the AAC task. In future work, we will explore the promotion of larger-scale data pre-training for AAC tasks, and try other effective methods to integrate similar audio captions information into AAC tasks.

## 6. 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] 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.

[3] Takeuchi, Daiki, et al. “Effects of word-frequency based pre- and post-processings for audio captioning,” in Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), 2020, pp. 190-195.

[4] Chen, Kun, et al. “Audio Captioning Based On Transformer And Pre-trained CNN,” in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020), 2020, pp. 21-26.

[5] Perez-Castanos, Sergi, et al., “Listen carefully and tell: an audio captioning system based on residual learning and gammatone audio representation,” in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020), 2020, pp. 150-154.

[6] Xu, Xuenan, et al., “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. 225-230.

[7] Gururangan S, Marasović A, Swayamdipta S, et al., “Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), 2020, pp. 8342-8360.

[8] 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,” in IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2020, vol. 28, pp. 2880-2894.

[9] Gemmeke, Jort F., et al., “Audio set: An ontology and human-labeled dataset for audio events,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 776-780.

[10] Yuan, Weiqiang, Han, Qichen, et al., “The DCASE 2021 Challenge Task 6 System: Automated Audio Captioning with Weakly Supervised Pre-training and Word Selection Methods,” DCASE2021 Challenge, Tech. Rep., 2021.

[11] Ko, Tom, et al. “Audio augmentation for speech recognition,” in Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH), 2015, pp. 3586-3589.

[12] Ko, Tom, et al., “A study on data augmentation of reverberant speech for robust speech recognition,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 5220-5224.

[13] 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.

[14] Chen, Q., Zhu, X., Ling, Z. H., Wei, S., Jiang, H., & Inkpen, D, “Enhanced LSTM for Natural Language Inference,” in Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2015, pp. 1657-1668.

[15] F. Font, G. Roma, and X. Serra, “Freesound technical demo,” in Int. Conf. Multimedia (MM’13), 2013, pp. 411-412.

[16] 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), 2019, pp. 139–143.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,”in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[18] 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.

[19] 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.

[20] 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,” in Proc. Interspeech 2019, pp. 2613–2617, 2019.

[21] 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.

[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] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in Text Summarization Branches Out. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[24] R. Vedantam, C. L. Zitnick, and D. Parikh, “Cider: Consensus-based image description evaluation,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 4566–4575.

[25] 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.

[26] 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.