A request for clarity over the End of Sequence token in the Self-Critical Sequence Training

Jia Cheng Hu[1][0009-0008-1611-966X], Roberto Cavicchioli[1][0000-0003-0166-0898], and Alessandro Capotondi[1][0000-0001-8705-0761]

University of Modena and Reggio Emilia
`name.surname@unimore.it`

Abstract. *The Image Captioning research field is currently compromised by the lack of transparency and awareness over the End-of-Sequence token (<Eos>) in the Self-Critical Sequence Training. If the <Eos> token is omitted, a model can boost its performance up to +4.1 CIDEr-D using trivial sentence fragments. While this phenomenon poses an obstacle to a fair evaluation and comparison of established works, people involved in new projects are given the arduous choice between lower scores and unsatisfactory descriptions due to the competitive nature of the research. This work proposes to solve the problem by spreading awareness of the issue itself. In particular, we invite future works to share a simple and informative signature with the help of a library called SacreEOS. Code available at* https://github.com/jchenghu/sacreeos.

1 Introduction

The standard training strategy of a modern Neural Image Captioning system includes a policy gradient method, called Self-Critical Sequence Training [27] (shortened as SCST) which is designed to maximize the evaluation score given to the outputs. In this work, we discuss the problems caused by the lack of transparency from the research community over the inclusion or omission of the End-of-Sequence token during the optimization. An easy-to-overlook implementation detail that can significantly increase the performance of any model despite yielding worse descriptions.

The lack of awareness of the impact of the End-of-Sequence (`<Eos>`) omission and the lack of explicit information on the SCST implementation during the reporting of results pose an obstacle to scientific progress as they make it challenging to compare established works and evaluate new ones. Our paper attempts to spread awareness about the issue and proposes a solution to increase transparency in future works. This paper is structured as follows: in Section 2, we discuss the problem of the End-of-Sequence omission and why it is a problem for the research community; in Section 3, we provide a qualitative and quantitative analysis of the issue and we sample some of the recent works in Image Captioning to demonstrate its pervasiveness and provide some practical examples of its impact; In Section 4, we propose a possible solution with the help of a Python library called SacreEOS; in Section 5, we mention some of the literature approaches, and, finally, we draw our conclusions in Section 6.

2 Problem Description

2.1 CIDEr Optimization

CIDEr [30] is an n-gram-based metric that evaluates the caption semantic content according to its similarities to the ground truths. Compared to the other metrics [24, 16, 3, 2], it exploits the entire corpus of reference descriptions in the attempt of backing the evaluation with the consensus of the majority of people. In particular, each n-gram w_k in sequence Z is weighted according to the *tf-idf* term $g^n_k(Z)$ defined as:

$$\frac{h^n_k(Z)}{\sum\limits_{w_l \in \Omega} h^n_l(Z)} \cdot log(\frac{|I|}{\sum\limits_{I_i \in I} \min(1, \sum\limits_{q} h^n_k(V^i_q))}) \tag{1}$$

where Ω is the set possible n-grams in the corpus, I is the set of corpus images and $h^n_k(Z)$, $h^n_k(V^i_j)$ represent the number of occurrences of n-gram w_k in the sequence Z and in the j-th ground truth of image $I_i \in I$. The CIDEr and its alternative (CIDEr-D), compute the similarity between the candidate and reference description as the number of matching n-grams, weighted according to Equation 1. We refer to [30] for additional details of the formula since they are unnecessary for the sake of the discussion.

The standard training practice of the Image Captioning model consists of a pre-training phase using the Cross-Entropy loss followed by a CIDer-D optimization by means of a policy gradient method called Self-Critical Sequence Training [27]. The latter minimizes the negative expected reward:

$$L_R(\theta) = -\mathbf{E}_{y_{1:T} \sim p_\theta}[r(y_{1:T})] \tag{2}$$

where r is the CIDEr function, and its gradient is approximated as follows:

$$\nabla_\theta L_R(\theta) \approx -(r(y^s_{1:T}) - r(y^b_{1:T}))\nabla_\theta log\ p_\theta(y^s_{1:T}) \tag{3}$$

where $y^s_{1:T}$ are the sampled captions and $y^b_{1:T}$ are the base predictions.

2.2 The End-of-Sequence token in SCST

Two properties are desirable in an image description: completeness and correctness. While the first goal is pursued by the reward maximization, the SCST algorithm provides no explicit control over the latter, which is instead implicitly encouraged by the sequentiality of the decoding process. A token predicted at a specific time step also determines the most likely n-grams in the following ones. Since all n-grams are extracted from linguistically correct references, the final description will be correct, at least locally. Unfortunately, the CIDEr score does not consider a sentence's global correctness, and this aspect can be easily exploited by the SCST if not carefully implemented. In particular, the algorithm is allowed to produce incomplete descriptions using trivial sentence fragments that almost certainly match some parts of any set of references. This is the reason

CIDEr-D: 125.8		CIDEr-D: 130.1
A herd of sheep are standing in a field	->	A herd of sheep standing in a field with a
A group of people standing under an umbrella	->	A group of people standing under an umbrella in the
A group of people riding on the back of elephants	->	A group of people riding on elephants in a
A dog standing next to a fence with a stuffed animal	->	A dog is standing next to a fence with a stuffed animal
A bunch of bananas in a box on a table	->	A bunch of bananas in a box with a
A man sitting on a motorcycle	->	A man sitting on a motorcycle in front of a
A person in a boat in the water	->	A boat in the water with mountains in the
An elephant walking down a dirt road	->	An elephant walking down a dirt road in a
A small bird sitting on top of a rock	->	A bird sitting on top of a rock in the
A brown cow laying on the side of a motorcycle	->	A brown cow laying on the side of a mtorcycle
A group of people standing at a market with fruit	->	A woman standing in a market with fruit on
A person walking down a street with a clock station	->	A person walking in an ally with a clock on

Fig. 1. Captions generated by the same model (the Transformer [29]) trained with different implementations of SCST on the MS-COCO [17] data set. (Left) The model is optimized by the standard SCST and achieves 125.8 CIDEr-D on the validation set. (Right) The model is optimized by an implementation of SCST in which the `<Eos>` token is omitted and achieves 130.1 CIDEr-D on the validation set.

why the standard SCST implementation includes the special End-of-Sequence token, abbreviated as `<Eos>`, in the definition of the n-grams space. With this precaution, the reward function encourages a correct sentence termination leveraging the fact that the *tf-idf* of the `<Eos>` token out-weights those of function words.

2.3 The problem of the `<Eos>` omission

The inclusion or exclusion of the `<Eos>` token in the SCST algorithm represents a small and easy-to-overlook detail that significantly impacts a captioning system's performance. In case the `<Eos>` token is omitted, the descriptions generated by the network are often terminated by trivial sentence fragments such as "`and a`", "`in the`", "`on top of`" and "`in front of`" (more examples in Figure 1).

However, despite the presence of artifacts, they achieve superior performances on popular benchmarks compared to the correct ones (Figure 1). In particular, the number of additional points yielded by the artifacts can even be greater than the range of values in which different models developed around the same period typically compete. Therefore, the Image Captioning research field is currently suffering from a lack of transparency and, in some cases lack of awareness over the importance of the `<Eos>` token in the SCST. The problem can be described from multiple perspectives:

– If details over the <Eos> token in the SCST implementation are unavailable, omitted, or simply overlooked, it becomes difficult to compare models in the literature fairly.
– Researchers that are aware of the issue are given the difficult choice between less competitive results and poorly formulated outputs.
– Finally, researchers that are not aware of the issue (especially the newcomers in the field of Image Captioning) are indirectly encouraged to adopt the implementations that generate compromised sentences because of their superior performances.

3 <Eos> Omission Impact Analysis

3.1 Experimental Setup

For the qualitative and quantitative analysis of artifacts we implement[1] the Transformer [29] with 3 layers, d_{model}=512 and d_{ff}=2048, trained on the COCO 2014 [17] data set using the Karpathy split [11]. The Faster-RCNN backbone provided by [1] is adopted. The learning procedure consists of a first training step on Cross Entropy loss for 8 epochs followed by the CIDEr-D optimization for 20 epochs. The following configurations are adopted:

1. batch size of 48, a learning rate of 2e-4 annealed by 0.8 every 2 epochs and warm-up of 10000 in case of Cross Entropy Loss;
2. batch size of 48, a learning rate of 1e-4 annealed by 0.8 every 2 epochs during the SCST.

Optimization details are provided only for the sake of reproducibility since the artifacts discussed in this work arise regardless of the architecture and optimization details. For the ensemble results, 4 model instances are generated with the aforementioned method differing only in the initialization seed. In the experiments, for each seed, the SCST in the Standard and No<Eos> configurations optimize the same pre-trained model.

3.2 Artifacts Analysis

The <Eos> token can be omitted in two aspects of SCST:

1. during the reward computation;
2. during the initialization of *tf-idfs*;

which leads to 4 implementation instances in case sampled descriptions are tokenized consistently with respect to the ground-truths. Table 1 reports the impact of each configuration over the final descriptions. Two cases are the focus of this work since most popular implementations fall into the (*tf-idf* Init. w/ <Eos>,

[1] code can be found in https://github.com/jchenghu/captioning_eos

	tf-idf Init. w/ `<Eos>`	*tf-idf* Init. w/o `<Eos>`
Reward w/ `<Eos>`	baseline score no artifacts	lower score with artifacts
Reward w/o `<Eos>`	lower score with artifacts	higher score with artifacts

Table 1. Impact of the `<Eos>` token in SCST over the final CIDEr-D score and outputs. “*tf-idf* Init." refers to the ground truth sentences involved in the calculation of document frequencies, and “Predictions" refers to the sampled predictions and respective references.

	Karpathy test split			Karpathy validation split		
	Standard	`No<Eos>` (ε) / δ	Cleaned / δ	Standard	`No<Eos>` (ε) / δ	Cleaned / δ
Seed 1	128.4	131.2 (48.3%) / +2.8	127.8 / -0.6	125.8	130.1 (47.5%) / +4.3	126.4 / +0.6
Seed 2	129.0	130.9 (49.3%) / +1.9	127.4 / -1.6	127.0	129.9 (48.1%) / +2.9	126.2 / -0.8
Seed 3	129.0	131.0 (50.3%) / +2.0	127.5 / -1.5	127.2	129.3 (47.6%) / +2.1	125.7 / -1.5
Seed 4	129.1	130.7 (50.4%) / +1.6	126.8 / -2.3	128.0	130.0 (50.6%) / +2.0	126.0 / -2.0
Avg	128.9	130.9 (49.6%) / +2.0	127.3 / -1.1	126.9	129.8 (48.6%) / +2.8	126.0 / -0.9
$\sum$	133.0	134.9 (50.2%) / +1.9	131.2 / -1.8	131.8	133.8 (49.5%) / +2.0	129.8 / -2.0

Table 2. Performance comparison the CIDEr-D optimization in Standard and `No<Eos>` training. “Cleaned" refers to the `No<Eos>` results but artifacts are removed prior to the evaluation. $\sum$ refers to the ensemble of the four models and ε represents the percentage of artifacts.

Prediction w/ `<Eos>`) and (*tf-idf* Init. w/o `<Eos>`, Prediction w/o `<Eos>`) configuration referred as “Standard" and “`No<Eos>`" respectively throughout the rest of this work.

In the `No<Eos>` configuration, results are affected by 8 classes of artifacts depending on how sequences are terminated, with the last token belonging to A={“`in`", “`a`", “`of`", “`the`", “`with`", “`on`", “`and`" “`*`"}, where “`*`" represents all the possible remaining cases. While all elements in the set A are just simplifications of longer trivial fragments such as “`and a`", “`in a`", “`with a`" and “`in front of`", the case of “`on`" may seem acceptable but the token is often part of uncommon formulations such as “`a beach with a surfboard on`" and “`a street with a bus on`". Nevertheless, “`on`" represents only a small fraction of all instances, which mostly end with the “`a`" token instead (see Figure 2.c).

Figure 2.a showcases the number of artifacts converging to 50% of the whole testing set as the number of epochs increases. Thus, both correct and compromised sentences are produced by the `<Eos>` omission, which means the network learns to inject the fragments following a non-trivial and unpredictable criteria for each sequence.

Figure 2.b and Table 3.2 showcase that a single model trained with SCST in the `No<Eos>` configuration consistently outperforms the standard one across all seeds, often by a large margin, with a maximum gain of +2.8 and +4.3

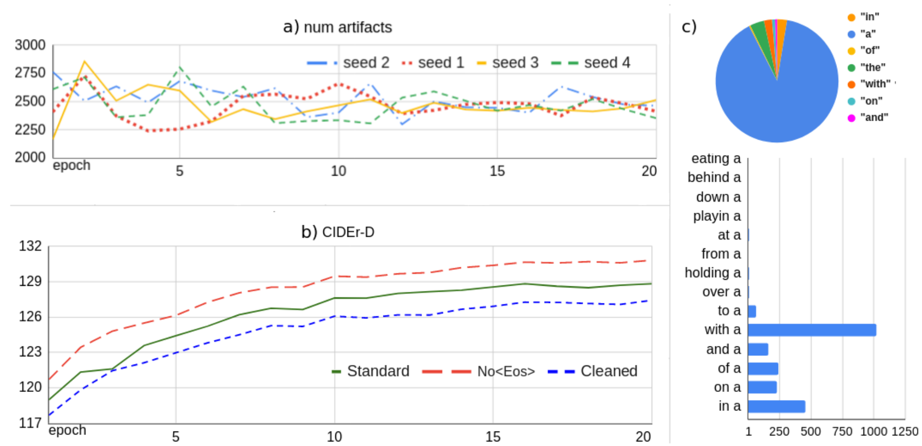

Fig. 2. a) The number of artifacts in the `No<Eos>` configuration on 5000 test set predictions. b) Average CIDEr-D score of 4 training instances (different seeds) in the Standard and `No<Eos>` configuration, "Cleaned" denotes the `No<Eos>` performance in case artifacts are removed before the evaluation. c) Artifacts distribution. Sequences terminated by "a" account for 89.8% of all cases (top). Histogram of sequences terminated by "a" (bottom).

CIDEr-D in the offline test and validation set respectively. Whereas, by removing the artifacts from the latter predictions we observed the opposite trend with a maximum performance decrease of -2.3 and -2.0. Therefore, the increase in score is mostly due to the artifacts and the `<Eos>` omission poses an obstacle to the generation of semantically meaningful content. Similar behaviour is observed for ensemble performances (referred as $\sum$).

3.3 Literature classification

We sample recent works in the research literature and classify each of them according to the way SCST is implemented. In Section 3.2 we observed that only half of the evaluated sentences are compromised, which means that if a paper provides only a few correct captioning examples, it is not enough to determine whether the `<Eos>` token was omitted or not. Because of that, the classification is made through code inspection. The classes and the respective criteria are defined as follows:

- Standard: `<Eos>` token is included in both SCST initialization and reward computation or complete results on either test or validation set are provided;
- `No<Eos>`: `<Eos>` token is omitted in both initialization and reward computation;
- `Unknown`: the code was not found or it was not available at the time this work was completed.

Table 3.3 showcases that only 12 of 25 works are confirmed to follow the Standard implementation, 8 fall in the `No<Eos>` category and 5 are unknown. The State-of-the-art architectures in 2019 [8] and 2020 [23] achieved 129.6 and 131.4 CIDEr-D scores respectively, which showcases the gradual improvement process of the research activity and provides an example of the magnitude of improvements over the years. Unfortunately, such a difference in performance can be lower than the additional score yielded by artifacts (see Section 3.2). For instance, if AoANet adopted the `No<Eos>` configuration, its score would have been comparable to the State-of-the-art performances of the following year (X-Transformer) (see Table 4).

The amount of `No<Eos>` implementations in the last years confirms the phenomena described in Section 2.3.

Year	Work	Offline	Online	SCST	Code inspection [a] (commit)
2018	GCN-LSTM [35]	127.6	-	`Unknown`	Code not found/available
2018	Up-Down [1]	120.1	120.5	Standard	peteanderson80/bottom-up-attention (514e561)
2019	HAN [32]	121.7	118.2	`Unknown`	Code not found/available.
2019	LBPF [26]	127.6	-	`Unknown`	Code not found/available
2019	RDN [12]	117.3	125.2	`Unknown`	Code not found/available
2019	SGAE [34]	127.8	-	Standard	yangxuntu/SGAE (af88115)
2019	Obj.Rel.Transf. [6]	128.3	-	Standard	yahoo/object_relation_transformer (6cf5bd8)
2019	AoANet [8]	129.8	129.6	Standard	husthuaan/AoANet (94ffe17)
2020	Ruotian Luo [20]	129.6	-	Standard	ruotianluo/self-critical.pytorch (be1a526)
2020	M^2 [5]	131.2	132.1	`No<Eos>`	aimagelab/meshed-memory-transformer (e0fe3fa)
2020	X-Transformer [23]	132.8	133.5	Standard	JDAI-CV/image-captioning (d39126d)
2020	Unified VLP [38]	129.3	-	Standard	LuoweiZhou/VLP (74c4d85)
2021	GET [9]	131.6	132.5	`No<Eos>`	luo3300612/image-captioning-DLCT (575b4dd)
2021	DLCT [21]	133.8	135.4	`No<Eos>`	luo3300612/image-captioning-DLCT (575b4dd)
2021	RSTNet [37]	135.6	134.0	`No<Eos>`	zhangxuying1004/RSTNet (e60715f)
2022	PureT [33]	138.2	138.3	Standard	232525/PureT (8dc9911)
2022	ExpansionNet [7]	140.4	140.8	Standard	jchenghu/ExpansionNet_v2 (365d130)
2022	BLIP [15]	136.7	-	Standard	salesforce/BLIP (3a29b74)
2022	CaMEL[4]	138.9	140.0	`No<Eos>`	aimagelab/camel (67cb062)
2022	GRIT [22]	144.2	143.8	`No<Eos>`	davidnvq/grit (32afb7e)
2022	S^2 [36]	133.5	135.0	`No<Eos>`	zchoi/S2-Transformer (c584e4)
2022	OFA [31]	154.9	149.6*	Standard	OFA-Sys/OFA (1809b55)
2022	ER-SAN [14]	135.3	-	Standard	CrossmodalGroup/ER-SAN (e80128d)
2022	CIIC [18]	133.1	129.2*	`Unknown`	Code not found/available
2022	Xmodal-Ctx [13]	139.9	-	`No<Eos>`	GT-RIPL/Xmodal-Ctx (d927eec)

Table 3. SCST classification of recent Image Captioning works and their respective performances on the MS-COCO 2014 task. The offline case reports the CIDEr-D score of a single model in contrast to the online evaluation server results where an ensemble is adopted instead with some exceptions denoted with "$\star$".

[a] Prefix https://github.com/

Model	RL epochs	Standard	No<Eos>	Ensemble	Set	δ
AoANet [8]	25	127.6	131.0	✗	test	+3.4
		126.2	130.3		val	+4.1
X-Transformer [23]	20	131.8	133.5	✗	test	+1.7
		130.1	132.3		val	+2.2
ExpansionNet [7]	12	143.7	145.3	✓	test	+1.6
		143.0	145.7		val	+2.7

Table 4. CIDEr-D performance increase observed in open source projects when the SCST configuration is changed from Standard into `No<Eos>` mode. Training details can be found in the respective works or repositories.

4 SacreEOS

4.1 SacreEOS signature

The lack of transparency and awareness over the `<Eos>` token in SCST originates from an easy-to-overlook implementation detail. Therefore, the natural solution is to disseminate awareness of the issue. To achieve this goal we introduce SacreEOS, a Python library whose main functionality consists of the generation of signatures that uniquely identify the key aspects of the SCST implementation. In particular, how the `<Eos>` token is handled. The sharing of the SacreEOS signature accomplishes three objectives:

1. it increases transparency and eases the comparison of models;
2. it informs the reader about the presence or absence of artifacts (those related to the `<Eos>` omission) in the results;
3. last but not least, it spreads awareness of the problem.

We believe this is especially useful in cases of works that do not release the code to the public.

Established researchers and existing implementations can manually generate the signature using the SacreEOS command line interface. The tool simply asks a few questions regarding the technical aspects of SCST, therefore it does not require any code integration. For new projects instead, SacreEOS consists of an SCST implementation helper, in this case, the signature is provided automatically. Format and signature examples are the following:

Format:

```
<scst config>_<Init>+<metric[args]>+<base[args]>+<Version>
```

Examples:

```
STANDARD_w/oInit+Cider-D[n5,s6.0]+average[nspi5]+1.0.0

NO<EOS>MODE_wInit+Cider-D[n4,s6.0]+greedy[nspi5]+1.0.0

NO<EOS>MODE_w/oInit+BLEU[n4]+average[nspi5]+1.0.0
```

4.2 Implementation helper and limitations of the approach

In addition to the functionality of signature generation, the SacreEOS library optionally provides helpful classes to ease the implementation of SCST in future projects. In particular, it covers the following aspects:

- *SCST class selection.* Given the number of established works implemented in both Standard and `No<Eos>` configurations, it is out of the scope of this paper to decide which one is the "correct" one (the library provides no default option in this regard). However, the tool helps the user to make informed decisions. Classes are currently defined by the reward metric, the reward base and whether the `<Eos>` token is included or omitted in both initialization and reward computation.
- *SCST initialization.* The library initializes the *tf-idfs* for the reward computation and performs input checks according to the selected class.
- *SCST reward computation.* The library currently supports the following reward functions CIDEr, CIDEr-D, CIDEr-R and BLEU. Results are consistent with the official repositories[2]. Each function is implemented in both Python and C, users can optionally enable the latter version to increase efficiency.
- *Signature generation.* In this case the SacreEOS signature is automatically determined by the class selection and does not require user intervention.

The library includes an intricate collection of assertions and input checks on all implementation levels, taylored to each specific class. Nevertheless, the SacreEOS does not prevent misreporting. In case the signature is manually generated, it relies on the user to provide the correct data.

5 Related works

The work of [27] mentioned the role of the End-of-Sequence token. However, it only provided a few qualitative examples and did not report numerical details. Several works in the past focused on improving the evaluation of Image Captioning systems but they mostly proposed alternatives to the CIDEr metric, such as TIGEr [10], SPIDEr [19], and CIDEr-R [28]. None of them addressed the issue discussed in this work.

The main inspiration of SacreEOS is SacreBLEU [25], in the field of Machine Translation, where ambiguities can arise from different tokenization and detokenization choices that ultimately affect the BLEU score [24].

6 Conclusion

Our work discussed the role of `<Eos>` in the Self-Critical Sequence Training and how the lack of transparency and awareness over its function pose an obstacle

[2] CIDEr, CIDEr-D, BLEU: github.com/vrama91/cider
CIDEr-R: github.com/gabrielsantosrv/coco-caption

to the scientific progress in the Image Captioning field. We described the source of the problem from a qualitative and quantitative perspective. We classified recent works in the scientific literature according to the SCST configuration to showcase the pervasiveness and the importance of the matter. Finally, we proposed a possible solution that consists of sharing a unique signature with the help of a Python library called SacreEOS, to enable fair model comparisons and spread awareness regarding the issue.

Bibliography

[1] Peter Anderson et al. "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.

[2] Peter Anderson et al. "Spice: Semantic propositional image caption evaluation". In: *European conference on computer vision*. Springer. 2016, pp. 382–398.

[3] Satanjeev Banerjee and Alon 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.

[4] Manuele Barraco et al. "CaMEL: Mean Teacher Learning for Image Captioning". In: *arXiv preprint arXiv:2202.10492* (2022).

[5] Marcella Cornia et al. "Meshed-memory transformer for image captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10578–10587.

[6] Simao Herdade et al. "Image captioning: Transforming objects into words". In: *Advances in Neural Information Processing Systems* 32 (2019).

[7] Jia Cheng Hu, Roberto Cavicchioli, and Alessandro Capotondi. "ExpansionNet v2: Block Static Expansion in fast end to end training for Image Captioning". In: *arXiv preprint arXiv:2208.06551* (2022).

[8] Lun Huang et al. "Attention on attention for image captioning". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 4634–4643.

[9] Jiayi Ji et al. "Improving image captioning by leveraging intra-and inter-layer global representation in transformer network". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 2. 2021, pp. 1655–1663.

[10] Ming Jiang et al. "Tiger: Text-to-image grounding for image caption evaluation". In: *arXiv preprint arXiv:1909.02050* (2019).

[11] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3128–3137.

[12] Lei Ke et al. "Reflective decoding network for image captioning". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 8888–8897.

[13] Chia-Wen Kuo and Zsolt Kira. "Beyond a Pre-Trained Object Detector: Cross-Modal Textual and Visual Context for Image Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 17969–17979.

[14] Jingyu Li et al. "ER-SAN: Enhanced-Adaptive Relation Self-Attention Network for Image Captioning". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 1081–1087.

[15] Junnan Li et al. "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation". In: *arXiv preprint arXiv:2201.12086* (2022).

[16] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[17] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[18] Bing Liu et al. "Show, Deconfound and Tell: Image Captioning With Causal Inference". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 18041–18050.

[19] Siqi Liu et al. "Improved image captioning via policy gradient optimization of spider". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 873–881.

[20] Ruotian Luo. "A better variant of self-critical sequence training". In: *arXiv preprint arXiv:2003.09971* (2020).

[21] Yunpeng Luo et al. "Dual-level collaborative transformer for image captioning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 3. 2021, pp. 2286–2293.

[22] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. "GRIT: Faster and Better Image captioning Transformer Using Dual Visual Features". In: *arXiv preprint arXiv:2207.09666* (2022).

[23] Yingwei Pan et al. "X-Linear Attention Networks for Image Captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 10971–10980.

[24] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[25] Matt Post. "A call for clarity in reporting BLEU scores". In: *arXiv preprint arXiv:1804.08771* (2018).

[26] Yu Qin et al. "Look back and predict forward in image captioning". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 8367–8375.

[27] Steven J Rennie et al. "Self-critical sequence training for image captioning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 7008–7024.

[28] Gabriel Oliveira dos Santos, Esther Luna Colombini, and Sandra Avila. "Cider-r: Robust consensus-based image description evaluation". In: *arXiv preprint arXiv:2109.13701* (2021).

[29] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[30] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. "Cider: Consensus-based image description evaluation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 4566–4575.

[31] Peng Wang et al. "Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework". In: *International Conference on Machine Learning*. PMLR. 2022, pp. 23318–23340.

[32] Weixuan Wang, Zhihong Chen, and Haifeng Hu. "Hierarchical attention network for image captioning". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 8957–8964.

[33] Yiyu Wang, Jungang Xu, and Yingfei Sun. "End-to-End Transformer Based Model for Image Captioning". In: *arXiv preprint arXiv:2203.15350* (2022).

[34] Xu Yang et al. "Auto-encoding scene graphs for image captioning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10685–10694.

[35] Ting Yao et al. "Exploring visual relationship for image captioning". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 684–699.

[36] Pengpeng Zeng et al. "S2 Transformer for Image Captioning". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*. Ed. by Lud De Raedt. Main Track. International Joint Conferences on Artificial Intelligence Organization, July 2022, pp. 1608–1614.

[37] Xuying Zhang et al. "RSTNet: Captioning with adaptive attention on visual and non-visual words". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 15465–15474.

[38] Luowei Zhou et al. "Unified vision-language pre-training for image captioning and vqa". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07. 2020, pp. 13041–13049.