

Fine-grained Image Captioning with CLIP Reward

Jaemin Cho¹ Seunghyun Yoon² Ajinkya Kale³ Franck Deroncourt²
Trung Bui² Mohit Bansal¹

¹UNC Chapel Hill ²Adobe Research ³Adobe Inc.

{jmincho, mbansal}@cs.unc.edu {syoon, akale, franck.deroncourt, bui}@adobe.com

Abstract

Modern image captioning models are usually trained with text similarity objectives. However, since reference captions in public datasets often describe the most salient common objects, models trained with the text similarity objectives tend to ignore specific and detailed aspects of an image that distinguish it from others. Towards more descriptive and distinctive caption generation, we propose to use CLIP, a multi-modal encoder trained on huge image-text pairs from the web, to calculate multi-modal similarity and use it as a reward function. We also propose a simple finetuning strategy of CLIP text encoder to improve grammar that does not require extra text annotation. This completely eliminates the need for reference captions during the reward computation. To comprehensively evaluate descriptive captions, we introduce FineCapEval, a new dataset for caption evaluation with fine-grained criteria: overall, background, object, relations. In our experiments on text-to-image retrieval and FineCapEval, the proposed CLIP-guided model generates more distinctive captions than the CIDEr-optimized model. We also show that our unsupervised grammar finetuning of the CLIP text encoder alleviates the degeneration problem of the naive CLIP reward. Lastly, we show human analysis where the annotators strongly prefer CLIP reward to CIDEr and MLE objectives on diverse criteria.¹

1 Introduction

Describing an image with its detailed, distinguishing aspects is crucial for many applications, such as creating text keys for image search engine and accessibility for the visual impaired. The standard deep learning approaches train an image-conditioned language model by maximizing the textual similarity between generated and reference

¹Code and Data: <https://github.com/j-min/CLIP-Caption-Reward>

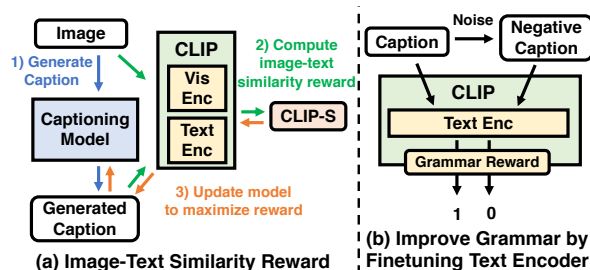


Figure 1: Overview of our proposed method. The left side illustrates our image captioning model training with image-text similarity reward based on CLIP (Sec. 3.1). The right side illustrates finetuning CLIP text encoder for improving grammar (Sec. 3.2).

captions (Vinyals et al., 2015; Xu et al., 2015; Rennie et al., 2017; Anderson et al., 2018). However, the reference captions of public datasets often describe only the most salient objects in images. This makes models trained to maximize textual similarity with reference captions tend to generate less distinctive captions that ignore the fine detailed aspects of an image that distinguishes it from others.

To alleviate the problem, we propose to use CLIP (Radford et al., 2021), a multi-modal encoder model trained on large image-text data (mostly English) collected from the web, by using its similarity scores (Hessel et al., 2021) as rewards (Sec. 3.1). In addition, we propose a CLIP text encoder finetuning strategy with synthetic negative caption augmentation to improve the grammar of captioning model, without any extra text annotations (Sec. 3.2). Note that our approach completely eliminates the need for reference captions during reward computation. To comprehensively evaluate descriptive captions, we also introduce FineCapEval, a new dataset that measures captioning in diverse aspects: overall, background, object, and relation between objects (Sec. 4).

In our experiments on MS COCO (Lin et al., 2014) dataset, we show that the captions from models trained with CLIP reward are more distinctive and contain more detailed information compared to

the captions from CIDEr (Vedantam et al., 2015)-optimized models. The CLIP-guided captions even achieve the higher text-to-image retrieval performance than reference captions that are originally paired with images. We also show that our text encoder finetuning significantly improves caption grammars by removing degeneration artifacts such as word repetition. In fine-grained caption evaluation with FineCapEval and human analysis, we show our CLIP based rewards outperform text similarity objectives by a large margin on all categories.

2 Related Works

Image Captioning Metrics. Traditionally, captions have been evaluated with n-gram or scene-graph based similarity metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), and SPICE (Anderson et al., 2016). However, such metrics often fail to capture paraphrased expressions due to the limited number of reference captions or scene-graphs. To tackle the problem, recent works including BERTScore (Zhang et al., 2019), ViLBERTScore (Lee et al., 2020a), UMIC (Lee et al., 2021), and CLIPScore (Hessel et al., 2021) propose to use relevance scores computed by language or multi-modal models pre-trained on large data.

Objectives for Image Captioning. Standard deep learning based image captioning approaches train models with maximum likelihood estimation (MLE) objective. Ranzato et al. (2016) point that MLE suffers from exposure bias problem.² To tackle exposure bias, Bengio et al. (2015) propose a curriculum learning strategy called scheduled sampling. Ranzato et al. (2016) propose to train models by directly maximizing the textual similarity between generated and reference captions with REINFORCE (Williams, 1992). Rennie et al. (2017); Luo (2020) propose self-critical sequence training (SCST) approach by normalizing rewards to stabilize its high variance.

Recent studies have observed that reference-trained captioning models often neglect important information from images (Dai et al., 2017; Wang et al., 2017). Lee et al. (2020b) use an visual question answering model’s accuracy as a reward, encouraging models to generate captions that in-

²While language models are trained with ground-truth previous context, they generate words based on the context words previously generated by themselves during inference.

clude information sufficient to answer a visual question. Dai and Lin (2017); Luo et al. (2018); Liu et al. (2018) use image-text retrieval model’s self-retrieval score as a reward and combine them with n-gram based metrics, encouraging captioning models to generate captions that are distinctive to each input image.

Note that these works require careful balancing between self-retrieval and text similarity objectives for stable training. In contrast, by finetuning CLIP text encoder (Sec. 3.2), our approach removes the need of reference caption and text similarity metrics for reward computation.

3 Methods

3.1 CLIP-guided Image Captioning

We propose to use the relevance score between image and text calculated by CLIP (Radford et al., 2021). Following Hessel et al. (2021), we use CLIP-S as our reward: $\text{CLIP-S}(I, c) = w * \max(\frac{f^I(I)^\top f^T(c)}{|f^I(I)| \cdot |f^T(c)|}, 0)$ where I, c are image and caption, f^I, f^T are CLIP’s image and text encoders, and w is set to 2.5. By maximizing the multimodal similarity of CLIP, which is a contrastively trained model, image captioning models are encouraged to generate captions that contain more distinctive information about the input image. Fig. 1 (a) illustrates this training strategy.

Following Rennie et al. (2017), we optimize our captioning model $P_\theta(c|I)$ with REINFORCE (Williams, 1992) with self-critical baseline. We approximate the gradient of the expected reward for generated caption \hat{c} , where rewards from beam search are normalized with the baseline rewards b from the greedy decoding \hat{c}_{greedy} : $\nabla_\theta \mathbb{E}_{\hat{c} \sim P_\theta(c|I)} [R(I, \hat{c})] \approx (R(I, \hat{c}_{beam}) - R(I, \hat{c}_{greedy})) \nabla_\theta \log P_\theta(\hat{c}_{beam}|I)$ where $R(I, c) = \text{CLIP-S}(I, c)$.

3.2 Improving Grammar with CLIP Text Encoder Finetuning

Since CLIP is not trained with a language modeling objective, the captioning model trained with CLIP-S reward often generates grammatically incorrect (e.g., repeated words) captions (See Table 3). We inject grammatical knowledge to CLIP’s text encoder with synthetic negative captions, generated by randomly repeating/removing/inserting/swapping/shuffling tokens of the reference captions. We provide the implementation details of such operations in appendix. We introduce a 2-layer per-


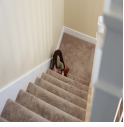
Image	Criteria	Annotations
(a) 	Background	white house, truck digging soil in front of the house, trees and bushes, house surrounded by a small garden, Mini excavator, houses, white and grey building, greenery, two houses, blue and white colored machine
	Object	a blue car, a blue car, black car, car, dozer, white and grey building, greenery, black car, green bushes
	Relation	parked in the front yard, in front, parked in front of, Parked, car standing on the road
	Overall	A blue car parked in the front yard of an off white house with a truck digging soil in front of the house. A blue car in front of a house surrounded by a small garden with trees and bushes in the background. A black car parked in front of a house with a mini excavator behind it with other houses in the background. A car and a dozer parked in front of two white and grey buildings and greenery on both sides. A black car standing on the road surrounded by green bushes on both sides and two houses and a blue and white colored machine in the background.
	Background	velvet carpet stairs, light-brown colored stairs, Off white wall, Cream painted walls, cream wall with straight line light
(b) 	Object	brown jumpsuit, kid, Toy, black jumpsuit, boy, brown clothes, toy, brown carpet, Little young boy, cotton carpeted stair, dark brown jumper dress, cream wall
	Relation	with its head on to, touching, Hiding, Holding, boy holding and playing with the toy, putting, wearing
	Overall	A child wearing a brown jumpsuit with its head on to the velvet carpet stairs. A kid is touching their head on a light brown colored stairs. A kid wearing a black jumpsuit and holding a toy hiding below the stairs with off white wall in the background. A boy wearing brown clothes holding and playing with his toy and playing on a brown carpet on stairs with cream painted walls. Little young boy is putting his forehead on the cotton carpeted stair wearing dark brown jumper dress and background of cream wall with straight line light.

Table 1: FineCapEval examples. For each image, we aggregate the annotations for each criteria from 5 different human annotators. For ‘overall’ criterion, we evaluate captions with CIDEr. For the rest of criteria, we evaluate captions with word-level recall R_{word} .

ception with sigmoid activation to CLIP text encoder’s feature $f^T(c)$, which outputs a grammar score $g(c) \in [0, 1]$, which is the probability of whether c is grammatically correct (reference) or not (negative). We train the parameters of the text encoder and grammar score predictor with CLIP’s original contrastive objective while fixing the image encoder parameters. Then we train the captioning models with the reward augmented with the grammar score: $R(I, c) = \text{CLIP-S}(I, c) + \lambda g(c)$, where $\lambda = 2.0$. We illustrate this finetuning strategy in Fig. 1 (b).

4 FineCapEval: Fine-grained Caption Evaluation Dataset

We introduce FineCapEval, a new dataset for caption evaluation in four different aspects. To construct FineCapEval, we collect 500 images from the MS COCO (Lin et al., 2014) test2015 split and Conceptual Caption (Sharma et al., 2018) val split, respectively. Then, for each image, we ask 5 human annotators to write phrases of 1) background, 2) objects (and their attributes; i.e., color, shape, etc.), 3) relation between objects (i.e., spatial relation), and 4) a detailed caption that includes all three aspects. See details of data collection process in appendix. In total, FineCapEval consists of 1,000 images with 5,000 annotations for each of the 4 criteria. In Table 1, we show samples of FineCapEval dataset.

5 Experiments

We train CLIP-Res50_{Transformer} captioning model (Shen et al., 2022) with different rewards: MLE, CIDEr, CLIP-S, CIDEr+CLIP-S, CLIP-

S+Grammar. Following previous works, we conduct experiment on MS COCO (Lin et al., 2014) English captioning dataset with Karpathy split (Karpathy and Fei-Fei, 2015). We evaluate the model with n-gram based metrics, embedding based metrics, text-to-image retrieval scores, and FineCapEval. We also conduct human evaluation with five criteria to understand the human preference of the generated captions in diverse aspects.

Model Architecture and Training. We use the CLIP-Res50_{Transformer} (Shen et al., 2022) as our captioning model architecture. The model consists of CLIP-Res50 for visual feature extraction and a transformer encoder-decoder for conditional language model. We resize images in 224x224 to extract 2048-dimensional visual features. The transformer consists of 6-layer encoder and 6-layer decoder. We train our the model with MLE objective for 15 epochs and further train with different rewards for 25 epochs (total 40 epochs), which takes within 1 day with 8 V100 GPUs. We use beam size 5 for beam search decoding. We implement a training pipeline with PyTorch (Paszke et al., 2017), PyTorch Lightning³, and HuggingFace Transformers (Wolf et al., 2020).

N-gram based Metrics. For N-gram based metrics, we report BLEU-4 (Papineni et al., 2002), CIDEr (Vedantam et al., 2015) METEOR (Banerjee and Lavie, 2005), and ROUGE-L (Lin, 2004).

Embedding-based Metrics. We report BERT-S (Zhang et al., 2019) and CLIP-S/RefCLIP-S (Hes-

³<https://github.com/PyTorchLightning/pytorch-lightning>

Model	Reward	N-gram based		Embed based				Text-to-Image Retrieval			FineCapEval				
		Text based				Image-Text based				Overall	Bg.	Obj.	Rel.		
		BLEU-4	CIDEr	METEOR	ROUGE-L	BERT-S	CLIP-S	RefCLIP-S	R@1					R@5	R@10
Ref. captions									29.5*	54.2*	65.0*				
CLIP-Res50	MLE	32.5	110.3	27.2	55.2	0.937	0.758	1.12	21.8	45.6	58.0	13.5	11.6	13.0	19.8
CLIP-Res50	CIDEr	38.2	124.9	28.7	58.5	0.942	0.759	1.13	20.9	45.6	58.2	12.8	13.1	23.1	22.4
CLIP-Res50	CLIP-S	6.2	11.2	18.7	31.6	0.882	0.860	1.17	42.5	71.6	82.2	13.9	20.8	26.4	24.9
CLIP-Res50	CIDEr+CLIP-S	37.7	124.6	28.8	58.3	0.941	0.772	1.14	24.4	50.2	63.1	13.0	13.0	23.4	21.7
CLIP-Res50	CLIP-S+Grammar	16.9	71.0	24.9	47.3	0.924	0.793	1.15	35.8	64.0	75.8	19.3	21.8	25.5	27.5

Table 2: Performance on MS COCO Karpathy test split. *The first caption out of 5 reference captions are used to calculate retrieval scores. R@K refers to the recall-K of the reference image. R_{word} refers to the word-level recall for background (Bg.), object (Obj.) and relation (Rel.) criteria (see Sec. 4 for details).

sel et al., 2021).⁴ BERT-S measures textual similarity between reference captions and generated captions, CLIP-S measures the image-text similarity between input images and generated captions, and RefCLIP-S averages the textual similarity (with reference captions) and image-text similarity.

Text-to-Image Retrieval. We report the recall of the reference image using a text-to-image retrieval model, to measure the distinctiveness of the generated captions. For the retrieval model, we use pretrained CLIP ViT-B/32 (Radford et al., 2021).

FineCapEval. For background, object, and relation criteria, we measure the captioning performance with word-level recall, $R_{word} \in [0, 1]$. See details of R_{word} calculation in appendix. For overall caption, we measure the performance with CIDEr.

Human Evaluation. To evaluate captions in terms of human preference, we show a pair of captions from CLIP-S+grammar reward (ours) with CIDEr reward and with MLE baseline to human annotators from Amazon Mechanical Turk⁵. Then we ask them to select a better caption on 5 criteria (overall, background, object, attribute, relation). For each of the 5 criteria, we ask 10 annotators with 50 pairwise selection questions. We use 50 images from FineCapEval for caption generation.

6 Results and Discussions

6.1 CLIP Guides Distinctive Captions

In Table 2, the models with CLIP-S and CLIP-S+Grammar rewards achieve higher image-text metrics (CLIP-S / RefCLIP-S) and text-to-image retrieval scores than baselines. Interestingly, their

⁴Following the default settings of original papers, BERT-S and CLIP-S/RefCLIP-S are based on RoBERTa-Large (Liu et al., 2019) and CLIP ViT-B/32 (Radford et al., 2021) respectively.

⁵<https://www.mturk.com/>

retrieval scores are even higher than the retrieval score with reference captions. This shows the distinctiveness of their generated captions. For image (a) in Table 3, our model with CLIP-S+Grammar reward describes the rainy weather with ‘wet’, while the model with CIDEr reward does not describe it.

Our models with CLIP-S and CLIP-S+Grammar rewards score lower text similarity metrics (n-gram based metrics and BERT-S) than the model with CIDEr reward. However, the low scores on these reference-based metrics can be addressed by that the models with CLIP-S and CLIP-S+Grammar rewards often generate captions that include fine-grained information that are not even present in the reference captions. For example, for image (b) in Table 3, CLIP-S+Grammar model describes ‘blue sign’ of the restaurant, whereas none of the reference captions mentions them.

6.2 Finetuning CLIP Text Encoder Improves Grammar

Table 3 shows that the degeneration (e.g., repeating words) of CLIP-S reward is successfully mitigated by adding the grammar reward (CLIP-S+Grammar). Table 2 shows that adding grammar reward significantly increases all text similarity metrics (e.g., +60 for CIDEr).

6.3 Fine-grained Caption Evaluation

FineCapEval. The rightmost four columns of Table 2 show that the captions with CLIP-S and CLIP-S+Grammar significantly outperforms the captions with CIDEr on all four criteria of FineCapEval: overall, background, object, relation. The gap is smallest in object criterion, which implies MS COCO reference captions describe more object information than background or relation between objects.

Human Evaluation. Table 4 shows human evaluation results on five criteria: overall, background,

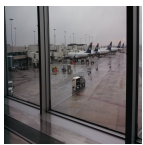

Image	Reward	Captions
(a) 	CIDEr	a window of an airport with planes on the runway
	CLIP-S	several rows of planes parked outside a terminal window area with fog outside a terminal window motion position area motion
	CLIP-S + Grammar	a lot of airplanes parked on a wet airport terminal
	Reference Captions	An airport filled with planes sitting on tarmacs. The view of runway from behind the windows of airport. a truck driving towards some planes parked on the runway Planes on a wet tarmac unloading at arrival gates. Window view from the inside of airplanes, baggage carrier and tarmac.
	CIDEr	a group of people riding bikes down a city street
(b) 	CLIP-S	several cyclists moving and bicycles near a restaurant and a blue advertisement outside a red brick building motion stance p
	CLIP-S + Grammar	a group of people riding their bikes on the busy street with a blue sign
	Reference Captions	people on bicycles ride down a busy street A group of people are riding bikes down the street in a bike lane bike riders passing Burger King in city street A group of bicyclists are riding in the bike lane. Bicyclists on a city street, most not using the bike lane

Table 3: Captions generated by models with different rewards on MS COCO Karpathy test split images.

Criteria	CLIP-S + Grammar	Win	Lose	Tie
Overall	v.s. MLE	49.0	41.8	9.2
	v.s. CIDEr	51.0	30.8	18.2
Background	v.s. MLE	52.8	35.0	12.2
	v.s. CIDEr	53.9	25.4	20.6
Object	v.s. MLE	52.0	36.6	11.4
	v.s. CIDEr	55.2	32.8	12.0
Attribute	v.s. MLE	57.2	36.8	6.0
	v.s. CIDEr	55.8	37.2	7.0
Relation	v.s. MLE	44.6	44.2	11.2
	v.s. CIDEr	49.2	39.6	11.2

Table 4: Human pairwise preference evaluation results.

object, attribute, relation. We sample 50 captions from model trained with CLIP-S+grammar reward (ours), CIDEr reward and MLE baseline using 50 images from Conceptual caption (Sharma et al., 2018) val split. For each of the 5 criteria, we ask 10 human annotators to select a better caption between ours and another method. On all criteria, the human annotators strongly prefer the captions with CLIP-S+Grammar rewards over CIDEr and MLE baseline.

7 Conclusion and Future Directions

We introduce a novel training strategy for image captioning models by maximizing multimodal similarity score of CLIP and finetuning its text encoder to improve grammar. The use of CLIP reward eliminates the need for reference captions and their bias for reward computation. We also introduce FineCapEval, a dataset for fine-grained caption evaluation. We demonstrate the effectiveness of our proposed method based on improvements in text-to-image retrieval, FineCapEval, and human evaluation on fine-grained criteria along with quali-

tative examples. Future works involve finetuning CLIP reward models with desired writing styles for different applications and improving the synthetic augmentation process by using external data suitable for grammars with advanced linguistics expertise.

8 Ethical Considerations

The CLIP models we used are trained on millions of image-text pairs collected from the web. Birhane et al. (2021) shows that such large-scale datasets often contain problematic and explicit image-text pairs. As the CLIP model card⁶ suggests, using CLIP reward for training image captioning models is intended as a research output, and any deployed use case of the models is out of scope.

Our captioning models and CLIP models are trained on English datasets; its use should be limited to English language use cases. As our proposed method is not limited to English and easily extended to other languages, future work will explore the extensions in various languages.

Acknowledgements

We thank the reviewers for their valuable comments. This work was partially done while JC was interning at Adobe Research and later extended at UNC, where it was supported by ARO Award W911NF2110220, DARPA MCS Grant N66001-19-2-4031, and NSF-CAREER Award 1846185. The views contained in this article are those of the authors and not of the funding agency.

⁶<https://github.com/openai/CLIP/blob/main/model-card.md>

References

- Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. [SPICE: Semantic Propositional Image Caption Evaluation](#). In *ECCV*.
- Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. [Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering](#). In *CVPR*.
- Satanjeev Banerjee and Alon Lavie. 2005. [METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments](#). In *ACL Workshop*.
- Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. [Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks](#). In *NIPS*, pages 1–9.
- Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. 2021. [Multimodal datasets: misogyny, pornography, and malignant stereotypes](#).
- Bo Dai, Sanja Fidler, Raquel Urtasun, and Dahua Lin. 2017. [Towards Diverse and Natural Image Descriptions via a Conditional GAN](#). In *ICCV*.
- Bo Dai and Dahua Lin. 2017. [Contrastive Learning for Image Captioning](#). In *NIPS*.
- Jack Hessel, Ari Holtzman, Maxwell Forbes, Roman Le Bras, and Yejin Choi. 2021. [CLIPScore: A Reference-free Evaluation Metric for Image Captioning](#). In *EMNLP*.
- Andrej Karpathy and Li Fei-Fei. 2015. [Deep Visual-Semantic Alignments for Generating Image Descriptions](#). In *CVPR*.
- Hwanhee Lee, Seunghyun Yoon, Franck Deroncourt, Trung Bui, and Kyomin Jung. 2021. [UMIC : An Unreferenced Metric for Image Captioning via Contrastive Learning](#). In *ACL*.
- Hwanhee Lee, Seunghyun Yoon, Franck Deroncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. 2020a. [ViLBERTScore: Evaluating Image Caption Using Vision-and-Language BERT](#). In *EMNLP Workshop*.
- Kenton Lee, Ming-wei Chang Jonathan, and H Clark Regina. 2020b. [CapWAP: Captioning with a Purpose](#). In *EMNLP*.
- Chin-Yew Lin. 2004. [ROUGE: A Package for Automatic Evaluation of Summaries](#). In *ACL Workshop*.
- Tsung Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. [Microsoft COCO: Common Objects in Context](#). In *ECCV*.
- Xihui Liu, Hongsheng Li, Jing Shao, Dapeng Chen, and Xiaogang Wang. 2018. [Show, tell and discriminate: Image captioning by self-retrieval with partially labeled data](#). In *ECCV*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#).
- Ruotian Luo. 2020. [A Better Variant of Self-Critical Sequence Training](#).
- Ruotian Luo, Gregory Shakhnarovich, Scott Cohen, and Brian Price. 2018. [Discriminability Objective for Training Descriptive Captions](#). In *CVPR*, pages 6964–6974.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wj Wei-jing Zhu. 2002. [BLEU: a Method for Automatic Evaluation of Machine Translation](#). In *ACL*.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chana, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. [Automatic differentiation in PyTorch](#). In *NIPS Workshop*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever, Jong Wook, Kim Chris, Hallacy Aditya, Ramesh Gabriel, Goh Sandhini, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. [Learning Transferable Visual Models From Natural Language Supervision](#). In *ICML*.
- Marc’Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. [Sequence Level Training with Recurrent Neural Networks](#). In *ICLR*, pages 1–15.
- Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. 2017. [Self-critical Sequence Training for Image Captioning](#). In *CVPR*.
- Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. [Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning](#). In *ACL*.
- Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. 2022. [How Much Can CLIP Benefit Vision-and-Language Tasks?](#) In *ICLR*.
- Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. [CIDEr: Consensus-based Image Description Evaluation](#). In *CVPR*.
- Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. [Show and Tell: A Neural Image Caption Generator](#). In *CVPR*.
- Liwei Wang, Alexander G. Schwing, and Svetlana Lazebnik. 2017. [Diverse and Accurate Image Description Using a Variational Auto-Encoder with an Additive Gaussian Encoding Space](#). In *NIPS*.

Ronald J. Williams. 1992. [Simple statistical gradient-following algorithms for connectionist reinforcement learning](#). *Machine Learning*, 8(3):229–256.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. [HuggingFace’s Transformers: State-of-the-art Natural Language Processing](#). In *EMNLP*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. [Show, Attend and Tell: Neural Image Caption Generation with Visual Attention](#). In *ICML*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. [BERTScore: Evaluating Text Generation with BERT](#). In *ICLR*.

In this appendix, we first show more example image captioning with different rewards (Sec. A). Then we explain the implementation details (Sec. B), and the details of FineCapEval (Sec. C). We also explain the details of human evaluation (Sec. D). Lastly, we provide the license for the datasets and models used in the project (Sec. E).

A More Image Captioning Examples

We provide more image captioning examples using different reward functions in Table 5. Overall, the captions from the model with CLIP-S+Grammar reward provide 1) more descriptive than the captions from the CIDEr model and reference captions, and 2) more grammatically correct than the captions from the model with CLIP-S reward.

B Implementation Details

Negative Caption Generation. In Alg. 1, we show Python implementation of the negative text generation (Sec. 3.2) for grammar finetuning. In summary, we generate negative captions using one of the operations: `repeat`, `remove`, `insert`, `swap`, `shuffle` on the original captions.

Evaluation Scripts. We use `pycocoevalcap`⁷ for MS COCO caption evaluation metrics such as CIDEr. We use BERTScore official repo⁸ with `roberta-large` model to calculate BERT-S. We report the evaluation script number from single run (single weight initialization), as we did not observe meaningful score fluctuation across multiple runs in our initial experiments.

⁷<https://github.com/tylin/coco-caption/tree/master/pycocoevalcap>

⁸https://github.com/Tiiiger/bert_score

C FineCapEval Details

Data Collection. To create a fine-grained description of the image, we ask annotators to write a caption that should describe target images’ 1) background, 2) objects and their attributes (i.e., color, shape, etc.), and 3) the relationship between the objects if any (i.e., spatial relation). Furthermore, we ask the annotators to write metadata containing which words/phrases in their writing belong to the three criteria. We also provide annotators with guidelines in writing a caption as follows: 1) There should be a single sentence describing the image. 2) The image may be a photo, an illustration or a pure background. 3) Pay close attention to local and global events in the image. 4) Descriptions should be at least ten words for each image. 5) Avoid the subject description of the image (i.e., a dog runs “very fast”, a man feels “successful”). 6) Avoid known entities such as specific locations (i.e. Eiffel Tower), time (i.e., 4 pm), event (i.e., Halloween), proper name. 7) In describing people, use only man/woman/boy/girl if clear; otherwise, use person/child. All annotators are hired by a professional crowdsourcing platform TELUS⁹. The crowdsourcing company obtained consents from the crowdworkers before the annotation process and conducted the ethical reviews. We collect English captions and all the annotators are native English speakers living in the US. We pay 5,400 USD, including 1) caption creation (5k samples) and 2) quality assurance process that manually examines 50% of the created caption by different workers.

Word-level Recall R_{word} . In Alg. 2, we show Python implementation of word-level recall R_{word} . In summary, R_{word} measures how many words from each of the reference phrases are included in a generated caption on average.

D Human Evaluation Details

We conduct pairwise evaluation of human preference, as shown in the Sec. 5. For each image, we show two captions generated from two models: ours (CLIP-S + Grammar) and the baseline (MLE/CIDEr). A human worker selects a caption that better describes the image in terms of five criteria: overall, background, object, attribute, and relation. For each criterion, we use 50 images from FineCapEval, and the two options are randomly and evenly shuffled. We also provide ‘Tie’ option

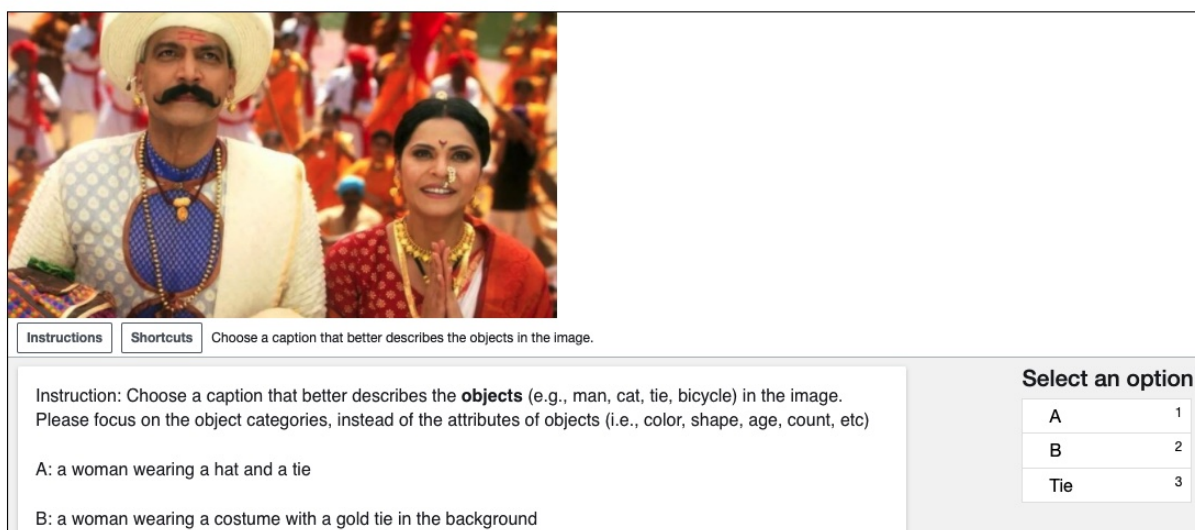
⁹<http://www.telusinternational.com>

Algorithm 1 Python implementation of negative text generation (main paper Sec. 3.2)

```
from random import randint, choice, shuffle
def repeat(tokens, n_max_gram=3, n_max_repeat=3): # repeat n-grams
    n_gram = randint(1, n_max_gram)
    repeat_idx = randint(0, len(tokens) - n_gram)
    repeated = tokens[repeat_idx:repeat_idx+n_gram]
    n_repeat = randint(1, n_max_repeat)
    for _ in range(n_repeat):
        insert_idx = randint(0, len(tokens))
        tokens = tokens[:insert_idx]+repeated+tokens[insert_idx:]
    return tokens
def remove(tokens, n_max_gram=3): # remove n-grams
    n_gram = randint(1, n_max_gram)
    remove_idx = randint(0, len(tokens) - n_gram)
    tokens = tokens[:remove_idx] + tokens[remove_idx + n_gram:]
    return tokens
def insert(tokens, vocab, n_max_tokens=3): # insert tokens
    n_insert_token = randint(1, n_max_tokens)
    for _ in range(n_insert_token):
        insert_idx = randint(0, len(tokens) - 1)
        insert_tok = choice(vocab)
        tokens = tokens[:insert_idx]+[insert_tok]+tokens[insert_idx:]
    return tokens
def swap(tokens, vocab, n_max_tokens=3): # swap tokens
    n_swap_tokens = randint(1, n_max_tokens)
    for _ in range(n_swap_tokens):
        swap_token_idx = randint(0, len(tokens) - 1)
        swap_token = choice(vocab)
        while swap_token == tokens[swap_token_idx]:
            swap_token = choice(vocab)
        tokens[swap_token_idx] = swap_token
    return tokens
def _shuffle(tokens): # shuffle tokens
    shuffle(tokens)
    return tokens
def generate_negative_text(text, vocab): # main function
    tokens = text.split()
    neg_type = choice(['repeat', 'remove', 'insert', 'swap', 'shuffle'])
    if neg_type == 'repeat': tokens = repeat(tokens)
    elif neg_type == 'remove': tokens = remove(tokens)
    elif neg_type == 'insert': tokens = insert(tokens, vocab)
    elif neg_type == 'swap': tokens = swap(tokens), vocab)
    elif neg_type == 'shuffle': tokens = _shuffle(tokens)
    return " ".join(tokens)
```

Algorithm 2 Python implementation of word-level recall R_{word} computation (main paper Sec. 5)

```
def calculate_word_recall(pred_id2sent, gt_id2phrases):  
    """  
    pred_id2sent: dict of generated captions (dict[int, str])  
    gt_id2phrases: dict of reference phrases (dict[int, list[str]])  
    """  
    n_total = 0  
    total_score = 0  
    for id, gt_phrases in gt_id2phrases.items():  
        pred_sent = pred_id2sent[id]  
        score = 0  
        for gt_phrase in gt_phrases:  
            word_score = 0  
            for gt_word in gt_phrase.split():  
                if gt_word in pred_sent:  
                    word_score += 1  
            score += word_score / len(gt_phrase.split())  
        score /= len(gt_phrases)  
        total_score += score  
        n_total += 1  
    word_recall = total_score / n_total * 100  
    return word_recall
```



Instructions Shortcuts Choose a caption that better describes the objects in the image.

Instruction: Choose a caption that better describes the **objects** (e.g., man, cat, tie, bicycle) in the image. Please focus on the object categories, instead of the attributes of objects (i.e., color, shape, age, count, etc)

A: a woman wearing a hat and a tie

B: a woman wearing a costume with a gold tie in the background

Select an option

A	1
B	2
Tie	3

Figure 2: The screenshot of human evaluation process for ‘object’ criterion (main paper Sec. 5).

Image	Reward	Captions
(a) 	CIDEr	a group of boats parked in the water on a lake
	CLIP-S	several rows of boats parked near a canal mountains horizon area and a mountain horizon horizon area horizon ear motion
	CLIP-S+Grammar	a lot of boats parked on the grass next to the lake with the hills behind
	Reference Captions	A blue boat docked on a green lush shore.
		A small marina with boats docked there
(b) 	CIDEr	a zebra standing in the snow next to a brick wall
	CLIP-S	a adult zebra wearing black and grey stripes standing near a brick wall area area with grey stance position stance
	CLIP-S+Grammar	a large black and grey zebra standing together in the snowy ground next to a stone
	Reference Captions	A zebra is standing outside in the snow
		One zebra standing in snow near a stone wall.
(c) 	CIDEr	a black dog sitting next to a plate of food
	CLIP-S	black black dog with macaroni and macaroni plate with pasta and pasta on a wooden floor plate position position position
	CLIP-S+Grammar	a black dog sitting next to a plate of food on the wood floor
	Reference Captions	Shaggy dog gets dinner served on a plate.
		A small black dog standing over a plate of food.
(d) 	CIDEr	two elephants standing next to a tree in a zoo
	CLIP-S	two adult adult and baby elephant near a tree enclosure area with a tree area enclosure motion stance ear stance
	CLIP-S+Grammar	a large elephant playing with a tree in the dirt field with rocks behind it
	Reference Captions	An elephant standing under the shade of a tree.
		An elephant standing in the middle of a rocky environment.
(e) 	CIDEr	a group of people riding bikes down a city street
	CLIP-S	several cyclists moving and bicycles near a restaurant and a blue advertisement outside a red brick building motion stance p
	CLIP-S+Grammar	a group of people riding their bikes on the busy street with a blue sign
	Reference Captions	people on bicycles ride down a busy street
		A group of people are riding bikes down the street in a bike lane
(f) 	CIDEr	a man riding a bike next to a train
	CLIP-S	older adult male riding a bicycle near a red and commuter train passing a train station motion stance ear stance
	CLIP-S+Grammar	a person walking on a bike next to a red passenger train on the road
	Reference Captions	A man on a bicycle riding next to a train
		A person is riding a bicycle but there is a train in the background.
(g) 	CIDEr	a window of an airport with planes on the runway
	CLIP-S	several rows of planes parked outside a terminal window area with fog outside a terminal window motion position area motion
	CLIP-S+Grammar	a lot of airplanes parked on a wet airport terminal
	Reference Captions	An airport filled with planes sitting on tarmacs.
		The view of runway from behind the windows of airport.

Table 5: More captions generated by models with different rewards on MS COCO Karpathy test split images.

to choose when the two captions are equally good or bad. For each criterion, we recruit 10 annotators 1) who are located in the Great Britain or the United States 2) HIT approval rate above 80% and 3) Number of HITs approved greater than 1000, from Amazon Mechanical Turk. We pay the annotators 0.03 USD per selection, which roughly corresponds to 11 USD/hour. In Fig. 2, we provide the screenshot for ‘object’ criterion for example.

E Licenses

For all artifacts, we remain within their respective license agreements. Here, we list the licenses:

- MS COCO - CC 4.0 - <https://cocodataset.org/#termsfuse>
- Conceptual Captions - <https://github.com/google-research-datasets/>

conceptual-captions/blob/
master/LICENSE

- **CLIP - MIT** - <https://github.com/openai/CLIP/blob/main/LICENSE>
- **CLIP-ViL - MIT** - <https://github.com/clip-vil/CLIP-ViL/blob/master/LICENSE>