Leveraging Partial Dependency Trees to Control Image Captions

Wenjie Zhong
The University of Tokyo
zvengin@is.s.u-tokyo.ac.jp

Yusuke Miyao
The University of Tokyo
yusuke@is.s.u-tokyo.ac.jp

Abstract

Controlling the generation of image captions attracts lots of attention recently. In this paper, we propose a framework leveraging *partial syntactic dependency trees* as control signals to make image captions include specified words and their syntactic structures. To achieve this purpose, we propose a ***S****yntactic* ***D****ependency* ***S****tructure* ***A****ware* ***M****odel* (**SDSAM**), which explicitly learns to generate the syntactic structures of image captions to include given partial dependency trees. In addition, we come up with a metric to evaluate how many specified words and their syntactic dependencies are included in generated captions. We carry out experiments on two standard datasets: Microsoft COCO and Flickr30k. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures. The code is available on GitHub[1].

1 Introduction

Controllable image captioning emerges as a popular research topic in recent years. Existing works attempt to enhance models' controllability and captions' diversity by controlling the attributes of image captions such as style (Mathews et al., 2016), sentiments (Gan et al., 2017), contents (Dai et al., 2018; Cornia et al., 2019; Zhong et al., 2020) and part-of-speech (Deshpande et al., 2019). However, some important attributes of image captions like words and syntactic structures, are ignored in previous works. For example, for the image in the Figure 2, the work (Cornia et al., 2019) specifies a set of objects like 'dog, man, frisebee' as a control signal, but there still exist lots of possibilities of composing them into different captions, such as 'a dog and a man play frisebee on grass' and 'a dog playing with a man catches frisebee', since both words and syntactic structures are not determined yet.

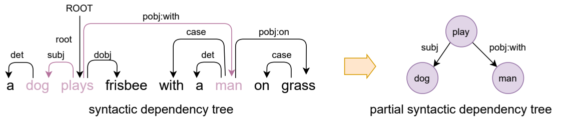
ROOT
pobj:with
case
pobj:on
root
det
subj
dobj
det
case
a dog plays frisbee with a man on grass
syntactic dependency tree
play
subj
pobj:with
dog
man
partial syntactic dependency tree

Figure 1: An example of syntactic dependency tree(left) and partial dependency tree (right)

To address this challenging issue, we propose a framework, which employs *partial dependency trees* as control signals. As shown in Figure 1, a partial dependency tree, a sub-tree of a syntactic dependency tree, contains words and their syntactic structures, and thus we can utilize it to specify control information about words and their syntactic structures.

In addition, we develop a pipeline model called *syntactic dependency structure-aware model* (SDSAM) which first derives a full syntactic dependency tree and then flatten it into a caption. The motivation behind this pipeline model is that we assume explicitly generating syntactic dependency trees as intermediate representations can better help the model learn how to apply the specified syntactic information to the captions and the intermediate representations can give users an intuitive impression on which part of the captions' syntactic structures is controlled.

Finally, we propose a syntactic dependency-based evaluation metric which evaluates whether the generated captions have been controlled in terms of syntactic structures. Our metric is computed based on the overlap of syntactic dependencies which is different from existing metrics like BLEU (Papineni et al., 2002), METEOR (Denkowski and Lavie, 2014), ROUGE (Lin, 2004), CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2018) which rely on the overlap of n-grams or semantic graphs. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures.

[1] https://github.com/ZVengin/DepControl_ALVR

Proceedings of the Second Workshop on Advances in Language and Vision Research, pages 16–21

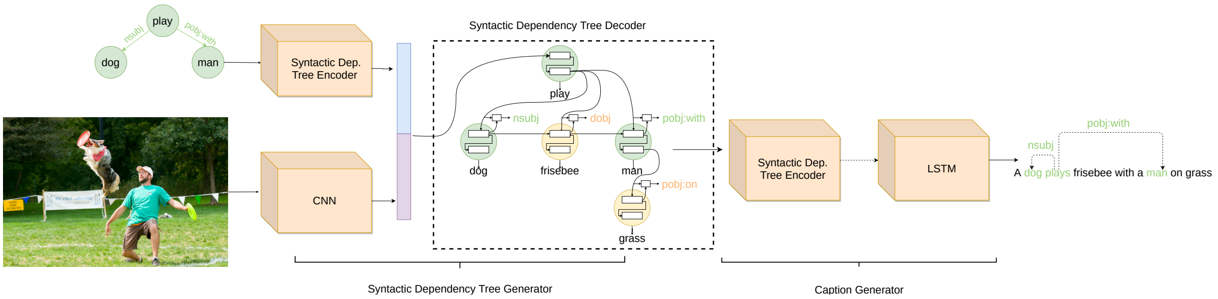
play
nsubj
pobj:with
dog
man
Syntactic Dep. Tree Encoder
CNN
Syntactic Dependency Tree Decoder
play
nsubj
dobj
pobj:with
dog
frisbee
man
pobj:on
grass
Syntactic Dep. Tree Encoder
LSTM
pobj:with
nsubj
A dog plays frisbee with a man on grass
Syntactic Dependency Tree Generator
Caption Generator

Figure 2: Model architecture: our model generates captions in two steps: (1) generating syntactic dependency tree using syntactic dependency tree generator. (2) flatting it into a caption using caption generator.

2 Framework Definition

The task presented in this paper is defined as generating a caption sentence (i.e. word sequence) $y = \langle w_1, \cdots, w_{|y|} \rangle$ given an image I and a partial dependency tree P as input, so that the dependency tree T_y of y includes P as far as possible. The syntactic dependency tree of a sentence, as shown in Figure 1, refers to a tree structure to represent syntactic relations between words. A syntactic dependency tree T_x of a sentence $x = \langle w_1, \cdots, w_{|x|} \rangle$ is defined as a set of dependencies, $\{D_1, D_2, \cdots, D_{|T_x|}\}$, where $|T_x|$ denotes the number of dependencies in T_x. Each dependency D_k is expressed in the form of $w_i \xrightarrow{e_{i,j}} w_j$, where w_i and w_j are the head word and the dependent word of D_k, and $e_{i,j}$ is the dependency label. We denote child nodes of w_i as $C(w_i)$; i.e. $C(w_i) = \{w_j | w_i \xrightarrow{e_{i,j}} w_j \in T_x\}$. A partial dependency tree P here refers to a sub-tree of the syntactic dependency tree of some sentence.That is, $P \subseteq T_x$ for some sentence x.

3 Syntactic Dependency Structure Aware Model

The syntactic dependency structure-aware model(SDSAM) shown in Figure 2 generates image captions in two steps: (1) the syntactic dependency tree generator on the left part derives a full syntactic dependency tree from the image and the partial dependency tree. (2) the caption generator on the right part will flatten the syntactic dependency tree into a caption.

The Syntactic Dependency Tree Generator The syntactic dependency tree generator encodes the input image with a CNN network implemented with Resent152 (He et al., 2016) into image features and encodes the partial dependency tree with a syntactic dependency tree encoder implemented with Tree-LSTM (Tai et al., 2015) into partial dependency tree features.

After combining the image features and the partial dependency tree features, the syntactic dependency tree generator derives the full syntactic dependency tree using the syntactic dependency tree decoder from the combined features $\boldsymbol{s}$. The syntactic dependency tree decoder consists of two attention modules, Attn_{in} and Attn_{out}, and two interweaved GRU networks (Cho et al., 2014), GRU_v and GRU_h. The decoding process is carried out from the root node to leaf nodes in a top-down manner. For a node w_i, its child nodes are decoded one by one from left-to-right. Each child node is predicted based on the information of its parent node and its left sibling node generated in previous steps. At the mean while, the attention modules highlight the words to be generated for the current child node. Assuming we decode the child w_j of node w_i, the hidden state of node w_i and node w_j are denoted as $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ respectively. The left sibling of node w_j is denoted as w_{j-1} and its hidden state as $\boldsymbol{h}_{j-1}$. For each input image, we detect a set of keywords $c = \{r_1, \cdots, r_{|c|}\}$ following the method proposed in (You et al., 2016), and encode c into a matrix $\boldsymbol{C} \in \mathbb{R}^{E_w \times |c|}$, where E_w is the size of word embedding.

$$\boldsymbol{h}_0 = \boldsymbol{U}^{(s)} \boldsymbol{s} \quad (1)$$

$$\tilde{\boldsymbol{h}}_i = \text{GRU}_v(\boldsymbol{h}_i, \boldsymbol{w}_i) \quad (2)$$

$$\boldsymbol{c}_{\text{in}} = \text{Attn}_{\text{in}}(\boldsymbol{w}_i, \boldsymbol{C}) \quad (3)$$

$$\boldsymbol{h}_j = \text{GRU}_h(\tilde{\boldsymbol{h}}_i, [\boldsymbol{h}_{j-1}; \boldsymbol{w}_{j-1}; \boldsymbol{c}_{\text{in}}]) \quad (4)$$

$$\boldsymbol{c}_{\text{out}} = \text{Attn}_{\text{out}}(\boldsymbol{h}_j, \boldsymbol{C}) \quad (5)$$

$$w_j \sim \text{Softmax}(\boldsymbol{U}^{(w)} \boldsymbol{h}_j + \boldsymbol{V}^{(w)} \boldsymbol{c}_{\text{out}}) \quad (6)$$

$$e_{i,j} \sim \text{Softmax}(\boldsymbol{U}^{(e)} \boldsymbol{h}_j + \boldsymbol{V}^{(e)} \tilde{\boldsymbol{h}}_i) \quad (7)$$

where:

$$\text{Attn}(\boldsymbol{q}, \boldsymbol{C}) = \boldsymbol{C}\boldsymbol{\alpha} \tag{8}$$

$$\boldsymbol{\alpha} = \text{Softmax}(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{v}) \tag{9}$$

$$\boldsymbol{A} = \tanh(\boldsymbol{U}^{(\alpha)}(\boldsymbol{q} \cdot \mathbf{1}^{\mathsf{T}}) + \boldsymbol{V}^{(\alpha)}\boldsymbol{C}) \tag{10}$$

In the above formulas, $\boldsymbol{U}^{(s)} \in \mathbb{R}^{H \times E_s}$, $\boldsymbol{U}^{(w)} \in \mathbb{R}^{V_w \times H}$, $\boldsymbol{U}^{(e)} \in \mathbb{R}^{V_e \times H}$,$\boldsymbol{U}^{(\alpha)} \in \mathbb{R}^{E_a \times E_q}$, $\boldsymbol{V}^{(w)} \in \mathbb{R}^{V_w \times H}$, $\boldsymbol{V}^{(e)} \in \mathbb{R}^{V_e \times H}$ and $\boldsymbol{V}^{(\alpha)} \in \mathbb{R}^{E_a \times E_w}$ are parameters for reshaping features. Here E_s, E_a and E_q are the size of the input feature $\boldsymbol{s}$, the attention feature $\boldsymbol{A}$ and the query $\boldsymbol{q}$ respectively. V_w and V_e are the vocabulary size for the node and edge respectively and H is the size of hidden states. In equation (10), $\boldsymbol{v} \in \mathbb{R}^{E_a \times 1}$ is a parameter and $\mathbf{1} \in \mathbb{R}^{|c| \times 1}$ is a vector with all elements being one.

The Caption Generator The caption generator takes the syntactic dependency tree generated in the first step as input and encodes it with the syntactic dependency tree encoder into syntactic dependency tree features. The caption generator combines it with image features extracted in the first step and use the combined features to initialize the LSTM decoder (Hochreiter and Schmidhuber, 1997) to generate the caption.

4 Experiment

Preparing Datasets with Partial Dependency Trees For evaluation, we apply two methods to create partial dependency trees for on Microsoft COCO (Chen et al., 2015) and Flickr30k (Young et al., 2014). The first method extracts partial dependency trees from reference captions. We parsing reference captions to syntactic dependency trees using Spacy [2] and then randomly sample subsets from each syntactic dependency tree. Sampled partial dependency trees are then paired with corresponding reference captions. The dataset created by this procedure is denoted as $test_{gold}$ in Section 5.

The other method creates partial dependency trees from images in two steps: (1) we first train a syntactic dependency classifier to predict syntactic dependencies for an input image. (2) Predicted syntactic dependencies are combined to form a syntactic dependency graph for the input image, from which partial dependency trees are sampled. The dataset created by this procedure is denoted as $test_{pred}$ in Section 5.

For training, following the first method, we directly sample a partial dependency tree from one of the reference captions for each image and the paired reference caption is used as a training target.

Evaluation Metric The evaluation metrics for image captioning fall into two categories: (1)Quality: evaluating the relevance to human annotations with metrics including BLEU (Papineni et al., 2002) and METEOR (Denkowski and Lavie, 2014); ROUGE (Lin, 2004), and CIDEr (Vedantam et al., 2015) and SPICE (Anderson et al., 2018). (2)Control-ability: evaluating whether generated image captions are successfully controlled by partial dependency trees. We devise a new metric called ***D**ependency **B**ased **E**valuation **M**etric* (**DBEM**) for this purpose. Assuming that a partial dependency tree $P = \{D_1, \cdots, D_{|P|}\}$ is input, DBEM calculates how many syntactic dependencies specified in the partial dependency tree are included in the dependency tree T_y of generated caption y. The DBEM score for the evaluation dataset is given as an average of this score for each input. Formally,

$$DBEM(P, T_y) = \frac{\sum_{D \in P} \mathbf{1}(D, T_y)}{|P|}, \tag{11}$$

$$\mathbf{1}(D, T) = \begin{cases} 1 & \text{if } D \in T \\ 0 & \text{if } D \notin T. \end{cases} \tag{12}$$

Experiment Setting The training of our model is split into two stages including training the syntactic dependency tree generator and training the caption generator. We set the size of hidden states to be 512, the word embedding size to be 512, and the dependency label embedding size to be 300. We train our model using the Adam optimizer (Kingma and Ba, 2015) with a learning rate $5e^{-4}$ for the first stage and $1e^{-4}$ for the second stage. Two models, including our SDSAM model and the NIC model (Vinyals et al., 2015) with its encoder being replaced with Resnet152, are compared under three different control inputs. (1) *None* control: input is an image. (2) *Half* control: input is an image and the words of a partial dependency tree. (3) *Full* control: input is an image and a partial dependency tree.

5 Results and Analysis

Quality (1) Results on $\text{test}_{\text{gold}}$: We show BLEU-4 (B4), METEOR (M), ROUGE (R), CIDEr (C), and SPICE (S) scores on $test_{gold}$ in Table 1, whose

[2] https://spacy.io

		Microsoft COCO					Flickr30k				
Control	Model	B-4	M	R	C	S	B-4	M	R	C	S
None	NIC	9.3	15.5	35.6	88.5	21.7	5.9	11.4	28.6	36.3	14.1
	SDSAM	9.6	16.0	35.5	94.4	23.7	4.6	10.8	25.8	34.9	14.8
Half	NIC	25.5	27.3	52.2	232.2	41.9	12.7	18.3	38.3	88.7	26.4
	SDSAM	24.7	26.6	52.6	234.4	44.1	12.4	18.4	40.2	103.8	32.8
Full	NIC	32.5	29.9	58.3	294.1	47.1	15.0	19.1	41.0	105.5	27.7
	SDSAM	30.2	29.2	57.1	282.3	48.4	13.4	18.9	41.5	114.2	33.7

Table 1: Evaluation of quality on $test_{gold}$. Each generated caption is only evaluated against its corresponding reference caption.

		Microsoft COCO					Flickr30k				
Control	Model	B-4	M	R	C	S	B-4	M	R	C	S
None	NIC	27.2	23.8	51.2	86.7	17.1	18.1	18.1	42.8	35.3	11.5
	SDSAM	28.0	24.5	50.9	90.2	18.0	15.8	17.5	39.7	35.6	11.7
Half	NIC	27.9	24.4	52.0	88.4	18.3	18.2	17.6	42.3	32.1	11.9
	SDSAM	26.8	24.4	51.1	88.7	18.5	17.2	17.4	40.5	32.9	12.3
Full	NIC	25.6	24.6	51.0	86.5	18.6	15.7	18.4	41.5	31.2	12.5
	SDSAM	26.0	24.5	50.8	87.7	19.0	16.7	17.7	40.7	32.4	12.4

Table 2: Evaluation of quality on $test_{pred}$. Each generated caption is evaluated against all reference captions of its corresponding image.

partial dependency trees are sampled from reference captions. This table shows that both NIC and SDSAM achieve significant improvements on evaluation scores when more control signals are input. This indicates that generated captions become closer to reference captions. These improvements are expectable since control signals contain information of reference captions. This result attests that partial dependency trees carry information useful for generating specific sentences. When both models are given the same control signals, SDSAM has comparable performance to NIC in n-gram based metrics (i.e. BLEU-4, METEOR, ROUGE and CIDEr), while achieving a significantly better performance on SPICE, which is a semantic relation based metric. This result reveals an interesting phenomenon that explicitly learning the syntactic structures of captions can improve performance on the semantic relation based metric.

(2) Results on $\text{test}_{\text{pred}}$: We show the evaluation results on $test_{pred}$ in Table 2, whose partial dependency trees are generated from images. For NIC and SDSAM, evaluation scores mostly remain the same level, but slight improvements are observed in SPICE. This result reveals that partial dependency trees generated from images do not have a significant impact on the quality of image captions, while giving partial dependency trees as control signals do not harm caption quality. For the same control signals, SDSAM has a better performance on SPICE in most cases, which follows the results on $test_{gold}$.

Controllability DBEM scores on $test_{gold}$ and $test_{pred}$ are shown in Table 3. The table shows that the DBEM scores of both models are very low when no control is given. This reveals that only a small proportion of syntactic dependencies in partial dependency trees appear in reference captions by chance, indicating that additional input to control syntactic structures is meaningful. When the models are given words as control signals, the DBEM scores are significantly increased, meaning that both models can infer syntactic structures from words even without explicit syntactic structure information. However, it is also clear that nearly half of the specified dependencies are missing in generated captions. These observations suggest that words provide useful information as control signals, but are insufficient to specify syntactic structures completely. When partial dependency trees are input, the DBEM scores further improve significantly. It means that most syntactic dependencies specified in partial dependency trees are included in generated captions. This result demonstrates that syntactic structure information plays an important role in precisely controlling image captions.

When the models are given no control signals, SDSAM has better DBEM scores than NIC. This is possibly because SDSAM explicitly learns to generate syntactic dependency trees, and can bet-

Control	Model	test$_{\text{gold}}$		test$_{\text{pred}}$	
		MSCOCO	Flickr30k	MSCOCO	Flickr30k
None	NIC	7.1	4.5	12.2	15.5
	SDSAM	9.5	5.4	19.6	19.8
Half	NIC	47.8	33.9	61.4	64.7
	SDSAM	51.4	44.6	64.2	72.3
Full	NIC	68.3	42.7	86.2	85.0
	SDSAM	69.5	52.9	87.1	87.5

Table 3: Evaluation of controllability (DBEM scores)

ter generate high-frequency syntactic dependencies that also frequently appear in partial dependency trees. When the models are given words and/or syntactic dependencies as control signals, SDSAM achieves higher DBEM scores than NIC. This result demonstrates that explicitly learning to generate syntactic dependency trees as an intermediate representation contributes to better controlling of image captions.

6 Case Study

In Figure 3, we show an example of the output from our model on *test$_{pred}$*. Our syntactic dependency classifier first predicts a syntactic dependency graph from the input image. Once the syntactic dependency graph is constructed, we sample three partial dependency trees with different node numbers as shown in the figure. Finally, our SDSAM model infers the captions from the input image and the partial dependency trees. From this example, it is obvious that all words and syntactic structures specified in partial dependency trees also appear in the generated captions. Furthermore, the three generated captions are considerably different from each other, demonstrating that giving partial dependency trees as control signals can improve captions' diversity.

7 Conclusion

We presented a framework for controlling image captions in terms of words and syntactic structures by giving partial dependency trees as control signals. We develop a syntactic dependency structure aware model to explicitly learn the syntactic structures in control signals. Empirical results show that image captions generated by our model are effectively controlled in terms of specified words and their syntactic structures. Furthermore, the results indicate that explicitly learning to generate the syntactic dependency trees of captions enhances the model's controllability.

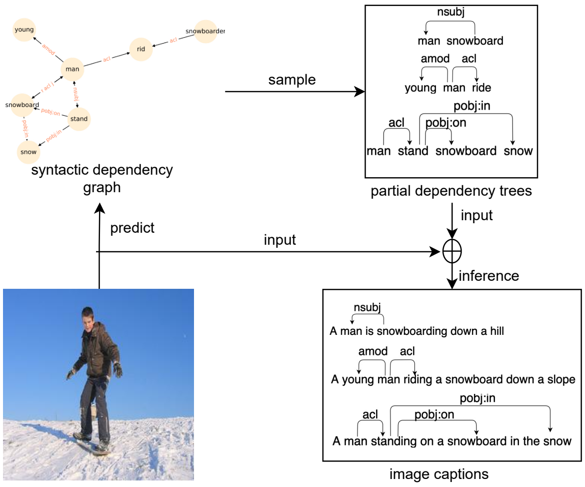

Figure 3: Case study: This figure shows an example generated during inference phase.

References

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 *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. 2015. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pages 1724–1734. Association for Computational Linguistics.

Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, control and tell: A framework for generating controllable and grounded captions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8307–8316. IEEE Computer Society.

Bo Dai, Sanja Fidler, and Dahua Lin. 2018. A neural compositional paradigm for image captioning. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 656–666.

Michael J. Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the Ninth Workshop on Statistical Machine Translation, WMT@ACL 2014, June 26-27, 2014, Baltimore, Maryland, USA*, pages 376–380. The Association for Computer Linguistics.

Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David A. Forsyth. 2019. Fast, diverse and accurate image captioning guided by part-of-speech. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10695–10704. IEEE Computer Society.

Chuang Gan, Zhe Gan, Xiaodong He, Jianfeng Gao, and Li Deng. 2017. Stylenet: Generating attractive visual captions with styles. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 955–964. IEEE Computer Society.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Alexander Patrick Mathews, Lexing Xie, and Xuming He. 2016. Senticap: Generating image descriptions with sentiments. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 3574–3580. AAAI Press.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1556–1566. The Association for Computer Linguistics.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4566–4575. IEEE Computer Society.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3156–3164. IEEE Computer Society.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image captioning with semantic attention. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 4651–4659. IEEE Computer Society.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Trans. Assoc. Comput. Linguistics*, 2:67–78.

Yiwu Zhong, Liwei Wang, Jianshu Chen, Dong Yu, and Yin Li. 2020. Comprehensive image captioning via scene graph decomposition. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIV*, volume 12359 of *Lecture Notes in Computer Science*, pages 211–229. Springer.