Generating Salient Scene Graphs with Weak Language Supervision

Alexandros Benetatos*, Markos Diomataris†, Vassilis Pitsikalis† and Petros Maragos*
*National Technical University of Athens (NTUA)
†deeplab.ai
https://github.com/deeplab-ai/SSGG-WLS

***Abstract*—Scene Graph Generation (SGG), given an image, is the task of building directed graphs where edges represent predicted `<subject - predicate - object>` triplets. Most SGG models struggle to identify important and descriptive relations in images flooding the graph with triplets like `<window - on - building>`. This is not due to training problems but rather the lack of saliency in fully supervised SGG datasets. Hence, observing that annotators describing an image naturally omit background relations and encode image saliency we (i) introduce a generalized method for training SGG models with weak supervision using image captions, (ii) introduce two variations of the Recall@N metric which can quantify the saliency of SGG models and (iii) perform quantitative and qualitative comparisons with related literature in VG200, where we achieve up to 35% improvement compared to re-implementation of the SOTA.**

***Index Terms*—scene graph generation (SGG), saliency, weak supervision, language supervision, image captions**

I. Introduction

Scene Graph Generation (SGG) has been tackled for years using fully supervised methods on VRD [1] and Visual Genome [2]. Unfortunately, the graphs produced from these models can not capture the required information to generate accurate interesting representations. Instead, models predict relations like `<sign - on - board>` (Fig. 1 - bottom) and `<window - on - building>` (Fig. 1 - top) that do not encode important information. As a result, these graphs can not be used to improve higher-level tasks like image captioning [3] where identifying important object relations is of the essence. We will refer to the ability of the models to identify important relations for the description of the image as saliency.

The problem begins with fully annotated SGG datasets. VG200 [4], the benchmark for SGG, is not exhaustively annotated as not all possible triplets or image entities are annotated. We also see that (i) relations not annotated are not always background or negatives (i.e. not true) (Fig. 1 - top) and (ii) many important entities are not annotated (Fig. 1 - bottom). As a result, models learn to label as background, relations that happened not to be annotated because of annotation bias.

Fortunately, there is an abundance of images with corresponding, easy-to-produce, captions incorporating saliency. Annotators describing an image naturally omit background relations (e.g. would not reference a window on a building when describing a scene of a man riding a horse). However, these datasets do not have grounding information, meaning that an entity in the text is not linked with an image region.

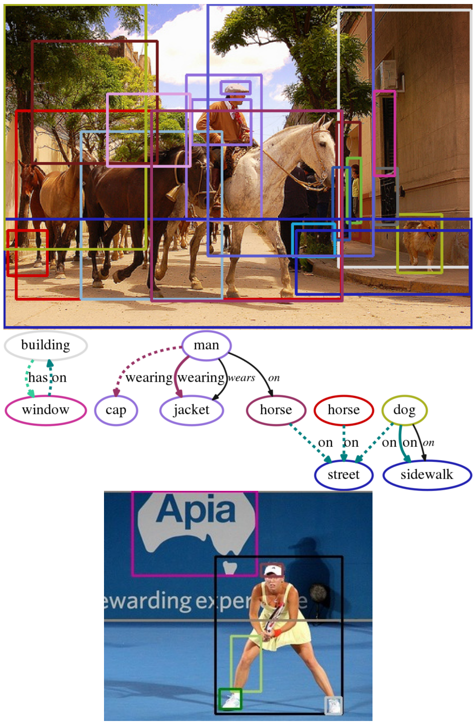

Fig. 1. (**top**) Even state-of-the-art methods [5] prefer relations like `<window - on - building` instead of `<man - riding - horse>`. Dotted arrows represent non-annotated but predicted pairs. Thin black lines show labeled annotations. (**bottom**) The racket, a main object, is not even annotated in this VG200 sample. Actually, the only annotated pair is `<sign - on - board>`.

We are the first to use weakly supervised training of SGG models on image captions to distill entity pairs' saliency. Additionally, we use weaker supervision than the previous state-of-the-art (SOTA) [6], despite improving test metrics by up to 35%. Lastly, our method is model-agnostic and can be used with other training loss modules like those in [7], [8].

Thus, we introduce new metrics to quantify SGG models' saliency, we implement a SOTA method for weak language supervision using image captions, we qualitatively and quantitatively examine the saliency of fully and weakly supervised models, and we compare with previous SOTA.

This research work was accomplished during A. Benetatos' diploma thesis, co-supervised by P. Maragos at NTUA, and M. Diomataris and V. Pitsikalis at Deeplab. This co-supervision is part of Deeplab's support of basic research to academic institutions/partnerships, and was funded as far as M. Diomataris, V. Pitsikalis, and any publication dissemination are concerned by Deeplab.

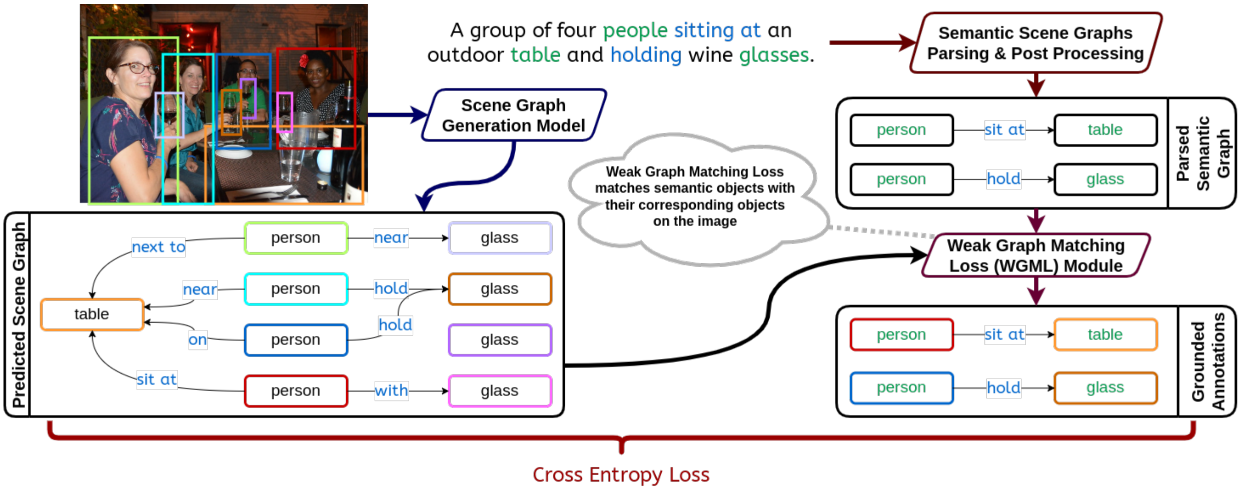
A group of four people sitting at an outdoor table and holding wine glasses.
Semantic Scene Graphs Parsing & Post Processing
Scene Graph Generation Model
Parsed Semantic Graph
person
sit at
table
person
hold
glass
Weak Graph Matching Loss matches semantic objects with their corresponding objects on the image
Weak Graph Matching Loss (WGML) Module
Predicted Scene Graph
person
near
glass
next to
near
person
hold
glass
table
on
person
hold
glass
sit at
person
with
glass
Grounded Annotations
person
sit at
table
person
hold
glass
Cross Entropy Loss

Fig. 2. Method Overview. We use an OTS Scene Graph Parser to extract weak supervision triplets from captions (Sec. III-A) and we detect the image entities using an OTS object detector (Sec. III-B). Lastly, we match the prediction with the supervision triplets, during training, using the Hungarian Algorithm to minimize a Weak Graph Matching Loss (WGML) (Sec III-C).

II. Related Work

Although relatively new, the idea of using image captions to train SGG models has been tested with mixed results [6], [9], [10] and no reference or indication of saliency. An off-the-shelf (OTS) scene graph parser is used to create semantic graphs from image captions and an OTS object detector to detect image entities. In [9], an iterative method like Expectation-Maximazation is used to approximate a Second Order Matching for semantic and detected objects. Predictions are made using visual and contextual language features from the semantic graphs, refined by a language LSTM. In [10] pre-trained Visual-Language transformers are used for grounding, creating a fully supervised dataset before training.

On the other hand, [6] randomly match semantic and visual objects of the same category to create a fully supervised dataset with soft labels. Then, a Vision-Language Transformer is trained in a fully supervised manner outperforming previous language-supervised SGG methods. A "weighted loss" term is also used which incorporates biases of the evaluation dataset into training. This improves results by more than 50% but we will ignore it as it leaks data from testing to training. Lastly, this is not a weakly supervised training method, as it offline creates a fully supervised dataset with soft labels.

In contrast, we post-process the parsed semantic graphs and detected objects and use a simple First Order Matching for semantic and detected objects during training.

III. Methods

Fully supervised datasets consist of `<subject - predicate - object>` triplets with bounding boxes grounding entities to images. In the weakly supervised unlocalized graph setting, we can use these datasets, ignoring image groundings, resulting in a set of triplets and image entities.

A weaker mode is to use only image captions as supervision. For this, we extract triplets from captions using OTS parsers (Sec. III-A) and detect objects using pre-trained object detectors (Sec. III-B). Thus, the problem reduces to the unlocalized graph setting with soft labels (i.e. we have machine-labeled triplets and image entity regions, without grounding between them). For training, we propose a Weak Graph Matching Loss (Sec. III-C) matching soft-labeled and predicted triplets.

A. Triplet extraction from image captions

To parse triplets from image captions, we use the OTS scene graph parser (SGP) provided by [11], where a syntactic dependency tree is built by [12] and then a rule-based method [13] is applied for transforming the tree to a scene graph.

We follow with post-processing to map the vocabulary generated by the SGP to that of VG200. For each lemmatized entity and predicate not in the VG200 vocabulary, we check if (i) one of their synonyms or hypernyms is in VG200, using WordNet [14], (ii) changing, singular to plural, adding or removing `ing` or changing tenses (e.g. from past to present), generates a word from VG200 and (iii) permuting between `on`, `onto`, `upon` and between `in`, `into`, `inside`, creates a word in VG200. If any check applies, we make the replacement.

B. Generate soft labels for image entities

To generate image entity labels, we could use a detector trained on the testing dataset. However, this would leak evaluation data into our training process. We use a pre-trained detector on the Open Images [15] dataset that detects 601 object categories, 77 common with VG200. We post-process the results similarly with section III-A, replacing entity class names and keeping only classes referenced on the triplets. Eventually, we can detect 109 of the 150 objects of VG200.

C. Weak Graph Matching Loss (WGML)

Next, we need to find a way to supervise each prediction of our model. For every predicted entity pair's relation we must know whether it should be foreground or not, and if so, which of the available soft labels should it be assigned. Let us consider figure 2 where we predicted <$\texttt{person}_2$ - holds - $\texttt{glass}_2$> and <$\texttt{person}_3$ - holds - $\texttt{glass}_2$>, but the soft labels inform us that <person - holds - glass>, not which person or glass. Our goal is to find which of the possible predicates between $\texttt{person}_{1-4}$ and $\texttt{glass}_{1-4}$ should be assigned the label holds and calculate the cross-entropy loss.

Our main contribution is implementing a simple First-Order Graph Matching using the Hungarian Algorithm [16] that calculates the label assignment minimizing the batch loss. Let G_i be the predicted image graph, G_s the semantic graph, M a graph matching connecting soft labels with inferred predicates, and L a loss function; we select the optimal matching as $M^* = \arg\min_M [\mathcal{L}(G_i, G_s, M)]$. Finally, if ϕ is the model parameters, the nested optimization that sums up the training procedure is $\phi^* = \arg\min_\phi \mathbb{E}[\min_M \mathcal{L}(G_i, G_s, M)]$.

The optimal matching is calculated by the Kuhn–Munkres (Hungarian) algorithm which, given the matching cost of each label-predicate pair, computes the minimum cost matching.

1) Triplet matching loss: Let i be a predicted triplet, j be a weakly labeled triplet, o_k, s_k, p_k be the entities and predicate category from triplet k, and $\mathbf{F}_{\mathbf{p_i}}$ be the distribution of the predicted predicates for triplet i, then the triplet loss is:

$$\mathcal{L}_{i,j} = \begin{cases} CE\left(\mathbf{F}_{\mathbf{p_i}}, p_j\right) & \forall i,j \; s.t. \; o_i = o_j \cap s_i = s_j \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

Hence, the Hungarian Algorithm, given the cost of $\mathcal{L}_{i,j} \forall i \in n_p, \forall j \in n_l$, where n_p, n_l is the number of predicted and labeled triplets accordingly, finds the optimal matching $M^* : \{0, 1, \ldots n_p\} \longrightarrow \{0, 1, \ldots n_l\}$ that minimizes the total cost:

$$\mathcal{L} = \sum_{i=0}^{n_p} CE\left(\mathbf{F}_{\mathbf{p_i}}, p_{M^*(i)}\right) \quad (2)$$

2) Independent predicate and saliency predictions: Adapting from [17], [18] we split the prediction of a relation S, P, O into two independent tasks (i) predict an S, O pair saliency score, $Pr(sl|S,O)$, and (ii) predict a predicate score, $Pr(P|S,O)$, as pairs can interact non-saliently. Hence, the probability entities S, O, are related with predicate P is:

$$Pr(P, sl|S,O) = Pr(P|S,O) Pr(sl|S,O) \quad (3)$$

IV. RESULTS

1) Introducing saliency metrics: To quantify the saliency of SGG models we introduce two new metrics; weak Recall@N for background predicate saliency (wR@N-bpsl) and for background saliency (wR@N-bsl). These evaluate the model's ability to choose salient triplets (bpsl) or entity pairs (bsl) by counting those in the top N predictions, that semantically match the weak annotations. Note these metrics are necessary, but not sufficient conditions for salient models (e.g. prediction of a wrong person holding the wrong glass for the annotation <person - holds - glass> would be evaluated as correct).

2) Training settings: We use COCO [19] captions and study our (Sec. III-A) and [6] parsing pipelines, and three different matching algorithms (Tab. I). The heuristic matches offline parsed triplets with entity pairs of high objectness scores.

TABLE I
ALL THE SETTING VARIATIONS FOR TRAINING THAT WE STUDY.

Setting Variations	Caption Preprocessing	Triplet Matching	Supervision
COCOSG weak	ours	HA (ours)	weak
SGGfromNLS weak	[6]		
COCOSG full best	ours	Heuristic (ours)	full w/ soft labels
SGGfromNLS full best	[6]		
COCOSG full random	ours	random [6]	full w/ soft labels
SGGfromNLS full random	[6]		

TABLE II
SIMPLE BASELINE USING ONLY LANGUAGE PRIORS.

Prior Probabilities	Object Detector Dataset	PredCls			SGGen		
		R@20	R@50	R@100	R@20	R@50	R@100
VG200	VG200 Open Images	41.940	56.861	63.586	8.664 5.107	12.202 7.460	14.912 9.284
COCOSG	VG200 Open Images	13.313	19.229	22.648	3.249 2.010	4.791 3.027	6.055 4.043
SGGFromNLS [6]	VG200 Open Images	10.755	15.683	18.490	2.855 1.898	4.149 2.776	5, 241 3.538

3) Training: Epoch time for VG200 on a 2080Ti GPU increases by 35% using WGML, while inference stays unchanged. We report results of best-out-of-two training times.

4) Priors baseline: We report the performance of a simplistic language model that, for each entity pair, predicts the most frequent predicate as computed from the training dataset (Tab. II). Note that our method for triplet extraction appears superior to [6] and the high sensitivity to different object detector evaluation or training datasets, makes the comparison between models and implementations challenging (Tab. II).

5) Unlocalized graph setting: Following [20], [21] we train on the unlocalized ground truth graph setting. We use VG200 and discard the grounding information. Using WGML and a simple model utilizing only linguistic and spatial information (LangSpat), we compare the weakly and fully supervised methods' and observe a performance decrease of just 1-2.5% (Tab. III), indicating our method's effectiveness.

6) Weak language supervision comparisons: We perform ablation using the UVTransE model [22] for different graph parsing and matching methods (Tab. I), and object detectors.

a) Graph parsing from captions: Our graph parsing method, COCOSG, leads to better evaluation results in VG200 compared to the previous SOTA, SGGFromNLS [6] (Tab. IV - top). This is attributed to different post-processing. For the saliency metrics (Tab. IV - bottom) the bpsl metric improves using our parsing, for all the graph matching methods. Still, the bsl metric, which only evaluates the saliency between pairs, does not similarly improve. This is because, independent of the triplet's post-processing steps, the information about the salient entities' categories already exists in the dataset.

b) Graph matching algorithms: Our dataset, COCOSG, using the WGML yields the best results on the VG200 (Tab. IV - top). However, for the dataset in [6] the heuristic matching performs better. This is attributed to increased noise on the labels of this dataset and the increased randomness during training. For the saliency metrics (Tab. IV - bottom) the bpsl metric prefers the models trained with the WGML. For the bsl metric, though, the heuristic seems to outperform our method.

TABLE III

RECALL@N COMPARISON IN VG200 FOR THE FULLY SUPERVISED AND WEAKLY SUPERVISED SETTING THAT DISCARDS THE GROUNDING INFORMATION FOR THE SEMANTIC ENTITIES IN THE TRIPLETS.

Method	Supervision	PredCls		
		R@20	R@50	R@100
LangSpat	Full	50.493	63.046	66.915
LangSpat + WGML	Weak	<u>47.981</u>	**61.558**	<u>65.586</u>
WSGM+IMP [4], [21]	Weak	**48.22**	<u>61.37</u>	**65.83**
VSPNet [20]	Weak	-	44.59	44.77

TABLE IV

COMPARISON OF DIFFERENT TRAINING SETTINGS (SEC. IV-2) BY COMBINING GRAPH PARSING AND MATCHING METHODS. (**TOP**) RECALL@N IN VG200. (**BOTTOM**) WEAK SALIENCY METRICS.

Dataset Variation	PredCls			SGGen		
	R@20	R@50	R@100	R@20	R@50	R@100
COCOSG weak (proposed)	**16.715**	**23.757**	**28.900**	**2.501**	**3.698**	**4.647**
SGGfromNLS [6] weak	10.008	15.008	19.066	2.003	2.790	3.446
COCOSG full best (ours)	<u>13.705</u>	<u>20.378</u>	<u>25.174</u>	<u>2.210</u>	<u>3.315</u>	<u>4.220</u>
SGGfromNLS [6] full best	11.274	17.253	21.820	2.094	2.904	3.669
COCOSG full random	10.135	15.846	20.085	1.656	2.635	3.438
SGGfromNLS full random [6]	8.970	14.625	19.320	1.860	2.675	3.359

Dataset Variation	SGGen					
	wR@5-bsl	wR@10-bsl	wR@20-bsl	wR@5-bpsl	wR@10-bpsl	wR@20-bpsl
COCOSG weak (proposed)	28.031	38.735	<u>48.292</u>	**15.425**	**21.522**	**28.215**
SGGfromNLS [6] weak	27.550	35.686	44.167	13.546	18.428	23.699
COCOSG full best (ours)	<u>29.154</u>	**39.033**	**49.370**	<u>15.081</u>	<u>20.032</u>	<u>25.762</u>
SGGfromNLS [6] full best	**30.117**	<u>38.872</u>	48.247	12.973	16.709	20.720
COCOSG full random	27.229	35.870	45.817	13.202	17.236	21.614
SGGfromNLS full random [6]	27.412	35.205	43.502	12.331	15.723	19.390

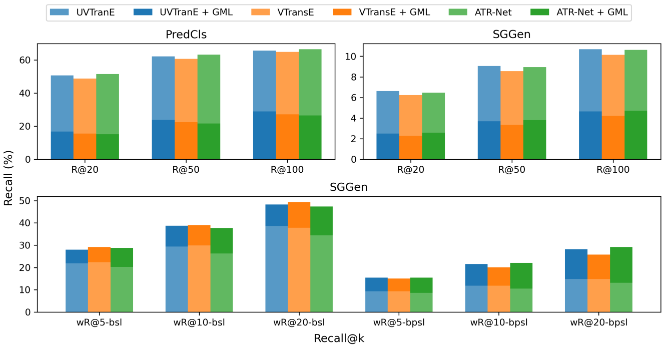

Fig. 3. Saliency (bottom) and VG200 (top) test results for the PredCls and SGGen setting from the re-implemented models. Training on the fully supervised VG200 and the weakly supervised COCOSG using the WGML. Weak language supervision greatly improves saliency.

TABLE V

RECALL@N COMPARISON IN VG200 (TOP) AND WEAK SALIENCY METRICS (BOTTOM) WITH DIFFERENT TRAINING DATASETS FOR OBJECT DETECTORS USED IN THE SGGEN SETTING EVALUATION. WE USE DETECTIONS GENERATED FROM A PRE-TRAINED MODEL ON "OPEN IMAGES" OR FROM ONE TRAINED ON VG200

Object Detector Finetuning Dataset		PredCls			SGGen		
for Training	for Inference (SGGen)	R@20	R@50	R@100	R@20	R@50	R@100
Open Images	Open Images	16.715	23.757	28.900	2.501	3.698	$4,647$
Open Images	VG200				3.330	5.070	6.621

Object Detector Finetuning Dataset		SGGen					
for Training	for Inference	wR@5-bsl	wR@10-bsl	wR@20-bsl	wR@5-bpsl	wR@10-bpsl	wR@20-bpsl
Open Images	Open Images	28.031	38.735	48.292	15.425	21.522	28.215
Open Images	VG200	19.574	27.252	34.999	11.59	16.800	22.026

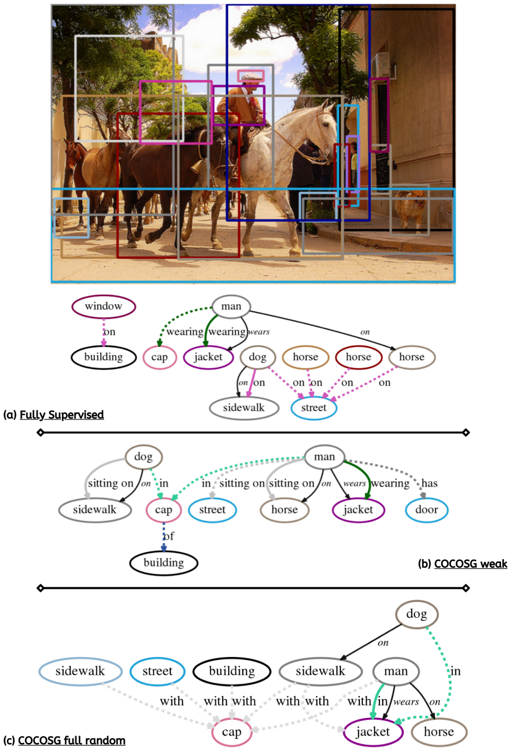

Fig. 4. Qualitative results for the effectiveness of our method (b) compared to [6] (c) and fully supervised training (a). Dotted arrows represent non-annotated object pairs. Thin black lines with italics show labeled annotations. Our method finds the main image relation `<man-sitting on-horse>` and selects a more descriptive predicate than "on" which is annotated, in contrast with the other methods that did not choose this interaction as important.

This is because, if the wrong visual and semantic entities are matched, the correct entities' categories are matched, tricking the bsl metric. Also, offline matching leads to less noise during training. Note the matching method from [6] struggles so much during training (last two rows of Tab. IV - top), that is outperformed by the language priors baseline (Tab. II).

c) Object detector: The choice of object detector and its training dataset can greatly impact models' performance (Tab. V). While an object detector trained on VG200 improves performance in VG200-related metrics, a detector trained on the more general "Open Images" improves the saliency metrics (Tab. V) and is not constrained to the VG200 dataset.

d) Applying WGML in multiple models: To examine its effectiveness and have comparable results, we apply WGML on re-implementations of VTransE [23], ATR-Net [17] and UVTransE [22]. Expectedly, the performance on VG200 reduces since the models are trained in a different dataset. Still, the models show a clear saliency improvement (Fig. 3), confirming that fully annotated datasets do not encoding it.

7) Qualitative results: We argue that our method improves scene graphs quality (Fig. 4, 5). The top eight predictions of each method are chosen to show the salient predictions. For every method, we use the UVTransE model.

a) More salient results using WGML: Our method (b) tends to detect the main relationship of the scene more easily and sometimes even selects more salient predicates than those annotated in VG200 (Fig. 4, 5). Many times,

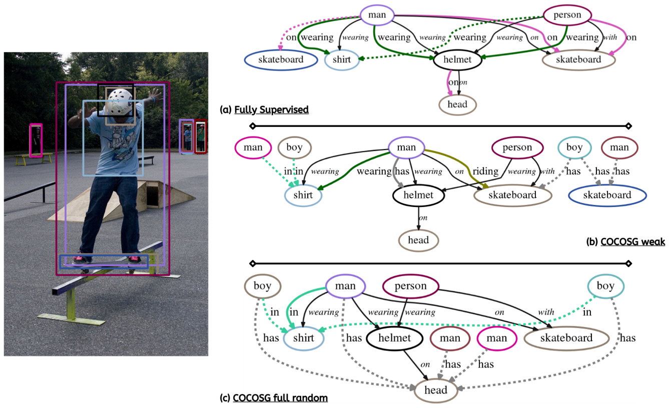

Fig. 5. Qualitative results for the effectiveness of our method (b) compared to [6] (c) and fully supervised training (a). Dotted arrows represent non-annotated object pairs. Thin black lines with italics show labeled annotations. Our method selects the most descriptive predicate `riding` to describe the relation between `man` and `skateboard` instead of the annotated `on`. In contrast, [6] does not detect that `skateboard` is important for the scene.

using the "random" method from [6] (c) the model can not detect the salient triplets in the image, and sometimes it has very little understanding of what is happening in the scene (Fig. 4, 5). The fully supervised training (a), many times struggles to detect the salient interactions in the image (Fig. 4), and even when it correctly detects salient pairs, it usually assigns generic non-salient predicates (Fig. 5). Unfortunately, our method's improved saliency, shown by predicting more interesting than annotated predicates (Fig. 4, 5), is punished by the Recall@N metric at VG200, highlighting, even more, the need for introducing the new metrics in Sec. IV-1.

b) Model failures: Sometimes our method predicts wrong triplets like `<cap-of-building>` (Fig. 4). The reason is the high frequency the predicate `of` is detected, by each-self, $Pr(P)$, and in specific entity context, $Pr(P|O)$, $Pr(P|S)$, $Pr(P|S,O)$ where P, S, O are the predicate, subject, and object. For our example, the model during training has seen many times the `<window/door-of-building>` and `<cap-of-woman/man/person>`, hence the predicate "of" ends up having very high probability. Thus, although the probability `cap` and `building` are related is low (seen from the low saliency score the model assigns to this pair), the triplet ends up in the top eight. More diverse vocabulary for entities and predicates, and variety in the caption datasets used will lead to a reduction of this problem. But, to evaluate on VG200, the choice of vocabulary was restricted for this work.

V. CONCLUSION

In this work, we focus on the lack of saliency in Scene Graph Generation (SGG) models, proposing metrics to quantify it and methods to produce more salient models. Fully annotated datasets are not salient, have unimportant relations annotated, and don't always refer to the main entities. Thus, observing that image captions encode which relations are more important in images, we parse semantic graphs from captions to weakly train SGG models. We introduce a First Order Matching for semantic and predicted graphs using Cross-Entropy and two variations of Recall@N that use weak supervision signals. Overall, our method generates more salient graphs, with descriptive predicates and salient entity pairs.

Future work will focus on removing the object detector and semantic graph parser, generating graphs end-to-end through attention distillation from captions with less supervision.

REFERENCES

[1] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual Relationship Detection with Language Priors," in *Proc. ECCV*, 2016.

[2] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016.

[3] V. Milewski, M. Moens, and I. Calixto, "Are scene graphs good enough to improve image captioning?," in *AACL*, 2020.

[4] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene Graph Generation by Iterative Message Passing," in *Proc. CVPR*, 2017.

[5] L. Mi and Z. Chen, "Hierarchical Graph Attention Network for Visual Relationship Detection," in *Proc. CVPR*, 2020.

[6] Y. Zhong, J. Shi, J. Yang, C. Xu, and Y. Li, "Learning to generate scene graph from natural language supervision," in *ICCV*, 2021.

[7] M. Diomataris, N. Gkanatsios, V. Pitsikalis, and P. Maragos, "Grounding consistency: Distilling spatial common sense for precise visual relationship detection," in *Proc. ICCV*, 2021, pp. 15891–15900.

[8] J. Zhang, K. J. Shih, A. Elgammal, A. Tao, and B. Catanzaro, "Graphical contrastive losses for scene graph parsing," in *CVPR*, 2019.

[9] K. Ye and A. Kovashka, "Linguistic structures as weak supervision for visual scene graph generation," in *Proc. CVPR*, June 2021.

[10] X. Li, L. Chen, W. Ma, Y. Yang, and J. Xiao, "Integrating object-aware and interaction-aware knowledge for weakly supervised scene graph generation," in *Proc. ACM Multimedia*. 2022, p. 4204–4213, ACM.

[11] P. Anderson, B. Fernando, M. Johnson, and S. Gould, "Spice: Semantic propositional image caption evaluation," in *ECCV*, 2016.

[12] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proc. ACL*, July 2003, pp. 423–430.

[13] S. Schuster, R. Krishna, A. Chang, L. Fei-Fei, and C. D. Manning, "Generating semantically precise scene graphs from textual descriptions for improved image retrieval," in *Workshop on Vision and Language (VL15)*. September 2015, ACL.

[14] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, nov 1995.

[15] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, and V. Gomes, "Openimages: A public dataset for large-scale multi-label and multi-class image classification.," https://storage.googleapis.com/openimages/web/index.html, 2017.

[16] H.W. Kuhn, "The hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, pp. 83–98, 01 1955.

[17] N. Gkanatsios, V. Pitsikalis, P. Koutras, and P. Maragos, "Attention-Translation-Relation Network for Scalable Scene Graph Generation," in *Proc. ICCV Workshops*, 2019.

[18] H. Zhang, Z. Kyaw, J. Yu, and S. Chang, "PPR-FCN: Weakly Supervised Visual Relation Detection via Parallel Pairwise R-FCN," in *Proc. ICCV*, 2017.

[19] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[20] A. Zareian, S. Karaman, and S. Chang, "Weakly supervised visual semantic parsing," in *Proc. CVPR*, June 2020.

[21] J. Shi, Y. Zhong, N. Xu, Y. Li, and C. Xu, "A simple baseline for weakly-supervised scene graph generation," in *Proc. ICCV*, 2021, pp. 16373–16382.

[22] Z. Hung, A. Mallya, and S. Lazebnik, "Contextual Translation Embedding for Visual Relationship Detection and Scene Graph Generation," *PAMI*, 2020.

[23] H. Zhang, Z. Kyaw, S. Chang, and T. Chua, "Visual Translation Embedding Network for Visual Relation Detection," in *Proc. CVPR*, 2017.