Scenes and Surroundings: Scene Graph Generation using Relation Transformer

Anonymous Authors[1]

Abstract

The identification of objects in an image, together with their mutual relationships as a scene graph, can lead to a deep understanding of image content. Despite the recent advancement in deep learning, the detection and labeling of visual object relationships remain a challenging task. In this work, a novel local-context aware relation transformer architecture has been proposed which also exploits complex global object to object and object to edge interactions. Our hierarchical multi-head attention based approach efficiently captures dependencies between objects and predicts contextual relationships. In comparison to state-of-the-art approaches, we have achieved an overall mean **4.85%** improvement and new benchmark across all the scene graph generation tasks on the Visual Genome dataset.

1. Introduction

A *scene graph* is a graphical representation of an image consisting of multiple entities and their relationships expressed in triple format like $\langle subject, predicate, object\rangle$. Objects in the scene become *nodes* in the graph, and a directed *edge* denotes a mutual relationship or predicate. E.g.: In Fig. 1, ‘Eye’,‘Hair’,‘Head’,‘Man’ are objects or nodes and their mutual relationships are described by the predicates ‘has’,‘on’.

Automated scene graph generation is executed in two steps: first, the objects present in the image are detected, and, second, likely predicates are derived. Current state-of-the-art object detection approaches have achieved very good performance in spatially locating objects in an image, while those for relation prediction are still in a nascent stage. To achieve state-of-the-art performance, it is important to consider context information (which could be both local or global context) and utilize this information to model dependencies between objects and predicates. An extracted scene graph can be used in many applications like visual question answering(Ghosh et al., 2019), image retrieval(Schuster et al., 2015), image captioning(Li et al., 2017).

The primary challenge involved in scene graph generation is to understand the role of each object in an image, and how objects are related and influenced by others in the context of the whole image. For example, in Fig. 1, the presence of nodes like ‘Eye ’, ‘Hair’, ‘Nose’, ‘Head’, indicate that these together describe a face. Additionally, node ‘Shirt’ implies that this is a face of a ‘Human’and not an animal. Node dependencies are also important for predicting an edge or a pairwise relation. Conversely, spatial and semantic co-occurrence also helps in identifying node classes. A subsequent challenge is to predict correct predicates describing the exact relationship between two objects.

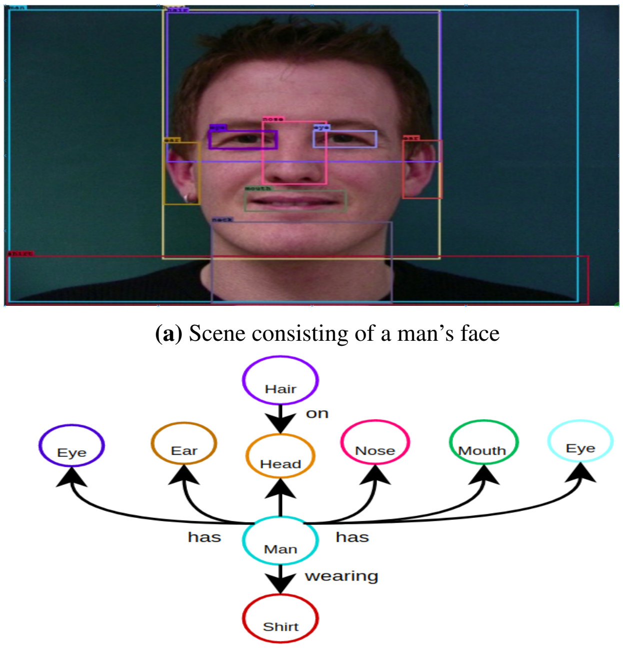
In this paper, we propose a novel scene graph generation architecture **Relation Transformer**, which leverages upon interactions among objects, predicates, their respective influence on each other as well as their co-occurrence patterns. Based on the above mentioned challenges, we have modified the Transformer architecture with some novel changes. To summarize our contributions:

- We have introduced a novel positional encoding algorithm for edges in the Transformer decoder that accumulates global scene context while preserving local context. This is specifically useful since an edge label can often be predicted from head or tail entity class.
- An algorithm that predicts an edge label needs to be aware of all node labels of other entities present in the scene as well as about other edge labels. To achieve this, we have applied an unrestricted attention and custom ordering of the E2N and E2E[0] blocks.
- We have achieved an overall mean 4.85% improvement over all scene graph generation tasks and set a new benchmark on the Visual Genome dataset.

2. Method

The scene graph generation task has been framed as a multi-hop attention based context propagation problem between

[0] According to the context in paper, we named encode-encoder to N2N(Node to Node), decoder-encoder to E2N(Edge to Node) and decoder-decoder to E2E(Edge to Edge) attention.



(a) Scene consisting of a man's face

(b) Corresponding scene graph

Figure 1. 1a is an example image of a face of a man. 1b describes the corresponding scene graph, annotated with various objects like head, ear, shirt (color coded as the respective bounding box) and their mutual relationships.

nodes and edges. This task is decomposed into four subtasks, starting with object detection, followed by modeling interactions between the nodes, then accumulating influence from both nodes and edges, and, finally, classifying relations between the objects. In the next sub-sections, we will describe these sub-tasks, along with a brief introduction of the attention mechanism, the Transformer, and their roles in these modules. An overview of the proposed Relation Transformer architecture is shown in Fig. 2.

2.1. Problem Decomposition

A scene graph $G = (N, E)$ of an image I is used for describing each node or object ($n_i \in N$) and their interlinked relations (like geometric, spatial etc.) with a directed edge ($e_{ij} \in E$). A set of nodes $\{n_i\}$, can be represented by their corresponding bounding boxes as B = $\{b_1, b_2, ..b_n\}$, $b_i \in \mathbb{R}^4$ and their class label O = $\{o_1, o_2..o_n\}$, $o_i \in C$. Each relation $r_{sub \to obj} \in R$ defines the relationship between the subject and object node. Hence, scene graph generation can be formulated as a three factor model as,

$$Pr(G|I) = Pr(B|I)\, Pr(O|B, I)\, Pr(R|O, B, I). \quad (1)$$

$Pr(B|I)$ can be inferred by any object detection model (Sec. 2.2). Sec.2.3.2 describes conditional probability of an object class $Pr(O|B, I)$, where the presence of one object can be influenced by another class. To model the relationships $Pr(R|O, B, I)$, we first compute an undirected edge (Sec. 2.3.3) between two objects, then conclude on a directed edge($r_{sub \to obj}$) (Sec. 2.4).

2.2. Object Detection

We have used Faster-RCNN (Ren et al., 2015) with a VGG-16 (Simonyan & Zisserman, 2014) backbone for object detection. For i^{th} object candidate, we obtain visual features $v_i^{RoI} \in \mathbb{R}^{4096}$, bounding box coordinates $b_i \in \mathbb{R}^5$ and class label probabilities[1] $o_i^{init} \in \mathbb{R}^{200}$. The initial feature ($n_i^{in} \in \mathbb{R}^{2048}$) of i^{th} node is obtained by applying a linear projection layer(f_{nlp}) on its concatenated features as described in Eq. 2. We have considered these individual proposals and their respective features as the initial node embeddings of the scene graph.

$$n_i^{in} = f_{nlp}([v_i^{RoI}, o_i^{init}, b_i]) \quad (2)$$

2.3. Context Propagation:

The core idea of our approach is the efficient context propagation across all nodes and edges using a Transformer encoder-decoder architecture (Vaswani et al., 2017). At the heart of the Transformer lies a self-attention mechanism, which is briefly described next.

2.3.1. Attention:

Attention mechanisms enable multi-hop information propagation in sequences ans graphs. The Transformer (Vaswani et al., 2017) architecture uses self-attention mechanisms for mapping of the global dependencies. One defines attention as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V. \quad (3)$$

The last equation describes a self-attention function, where query(Q), keys(K), and values(V) are a set of learnable matrices, and d_k is the scaling factor. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by multiplying a query matrix with its corresponding key.

2.3.2. Context Propagation for Objects:

Contextualization of objects not only enhances object detection (Liu et al., 2018) by exploring the surroundings of objects, but also encodes more expressive features for relation classification. For i^{th} node, we have used initial features from Eq. 2 along with a positional ($pos\_enc^n \in \mathbb{R}^{2048}$)

[1] GloVe embeddings for all classes has been used with a dimension of 200.

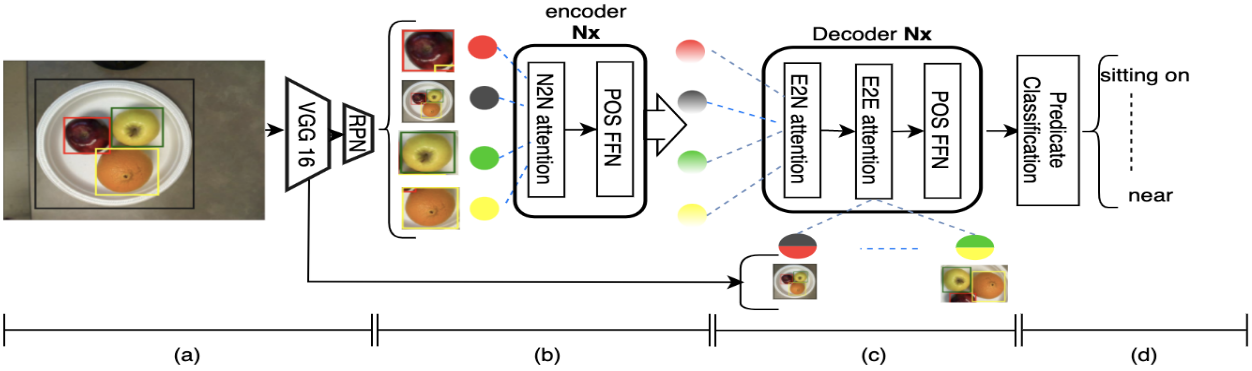


Figure 2. An overview of the proposed Relation Transformer architecture. The network consists of four stages: a) Feature generation by an object detector and bounding box extraction using RPN, b) Creation of context-rich node embeddings (light color) using N2N attention from initial nodes (dark color) c) Creation of edge embedding (bicolor based on respective nodes) using context from all nodes (E2N) and then from other edges (E2E), d) Classification of the relation using $\langle subject, edge, objects\rangle$ manner. Best viewed in color.

feature vectors, based on the actual position of the node in the sequence,

$$n_i^{final} = \text{encoder}(n_i^{in} + pos\_enc^{n}(n_i^{in})). \tag{4}$$

$$o_i^{final} = \text{argmax}(f_{classifier}(n_i^{final})). \tag{5}$$

After contextualization of the nodes by the encoder[2] in Eq. 4, we have obtained final node features (n_i^{final}). Final node features are subsequently used for two purposes. Firstly, they are passed to a linear object classifier (Eq. 5) to get the final object class ($o_i^{final} \in C$) probability and finally, the same node features are passed to the next module for edge context propagation.

2.3.3. Context Propagation for Edges

In this module, edge features are captured by accumulating context information across all nodes and edges. Edges are highly dependent on the local context, as they are associated with only a pair of nodes (subject, object). We have introduced novel changes in decoder, such that the network learns relational(E.g. spatial, semantic) influences from other nodes or edges by exploiting both local and global contexts.

For an edge belonging to i^{th} and j^{th}node, visual features $e_{ij}^{vis} \in \mathbb{R}^{4096}$ consist of the union of two object boxes $b_{i,j}$ as shown in Figure 2. Afterwards, spatial features $b_{i,j} \in \mathbb{R}^5$(b_i and b_j) are added with the concatenated GloVe (Pennington et al., 2014) embedding (e_{ij}^{sem}) of both classes. Subsequently, a linear projection layer (f_{elp}) is used to obtain the initial edge embeddings ($e_{i,j}^{in} \in \mathbb{R}^{2048}$) as

$$e_{i,j}^{in} = f_{elp}(e_{ij}^{vis} + b_{ij} + e_{ij}^{sem}) \tag{6}$$

As mentioned earlier, we have introduced three modifications in the Transformer decoder network such that it models the interaction between nodes and edges efficiently.

1. The decoder masked attention has been removed so that it can attend to the whole sequence, not just part of it.
2. A novel positional encoding vector has been introduced ($pos\_enc^{e_{ij}} \in \mathbb{R}^{2048}$) for edges ($e_{i,j}^{in}$) that encodes the position of both the source nodes, instead of the position of the edge alone. We hypothesize that it will be beneficial for the network to distinguish the source nodes (subject and object) out of all distinct nodes and the corresponding edges between source nodes to the other edges. This design bias can accumulate the global context without losing its focus on the local context or source nodes.

$$\begin{aligned} pos\_enc^{e_{ij}}_{(k,k+1)} &= [\sin(p_i/m^{2k/d_{dim}}), \cos(p_i/m^{2k/d_{dim}})]. \\ pos\_enc^{e_{ij}}_{(k+2,k+3)} &= [\sin(p_j/m^{2k/d_{dim}}), \cos(p_j/m^{2k/d_{dim}})]. \end{aligned} \tag{7}$$

   Eq. 7 describes positional encoding for an edge, where p_i and p_j are the positions of the nodes n_i and n_j, m is maximum number of sequence, $d_{dim} \in \mathbb{R}^{2048}$ is same dimension as $e_{i,j}^{in}$, and k denotes the k^{th} position in the positional encoding features vector.

3. The order of self-attention applied in the decoder has been altered. At first, E2N self-attention has been applied from an edge to all the nodes. Then, E2E attention from an edge to all the edges has been incorporated. Since, the edge is created from only two nodes, E2N attention accumulates necessary global context from all nodes. Afterwards, E2E attention will

[2] our encoder block remains same as Transformer, and its architecture shown in Figure 2.

Model	Graph constraint						No graph constraint				
	SGCLS			**PRDCLS**			**SGCLS**		**PRDCLS**		**Mean**
Recall@	20	50	100	20	50	100	50	100	50	100	
Message Passing (Xu et al., 2017)	31.7	34.6	35.4	52.7	59.3	61.3	43.4	47.2	75.2	83.6	52.44
Associative Embedding (Xu et al., 2017)	18.2	21.8	22.6	47.9	54.1	55.4	26.5	30.0	68.0	75.2	41.17
MotifNet(Left to Right) (Zellers et al., 2018)	32.9	35.8	36.5	58.5	65.2	67.1	44.5	47.7	81.1	88.3	55.76
Large Scale VRU (Zhang et al., 2019a)	36.0	36.7	36.7	66.8	68.4	68.4	-	-	-	-	52.16
ReIDN (Zhang et al., 2019b)	36.1	36.8	36.8	66.9	68.4	68.4	48.9	50.8	93.8	97.8	60.49
Relation Transformer (Ours)	**43.4**	**43.6**	**43.7**	**68.1**	**68.5**	**68.5**	**60.6**	**61.7**	**96.5**	**98.8**	**65.34**

Table 1. Comparison of our model with state of the art methods tested in Visual Genome (Krishna et al., 2017)

help an edge, enriched with global context, to learn from edges with similar relational embedding. Finally, we get contextual edge features ($e_{i,j}^{final} \in \mathbb{R}^{2048}$) as,

$$e_{i,j}^{final} = \text{decoder}(e_{i,j}^{in} + pos\_enc^{e_{ij}}) \tag{8}$$

2.4. Relation Classification

A relation is a directional property, i.e., *subject* and *object* cannot be exchanged. After obtaining the context-riched node and edge embeddings, a joint relational embedding ($rel_{emb} \in \mathbb{R}^{2048}$) has been created consisting of triplets like $\langle subject, edge, object \rangle$ followed by a Leaky ReLU (Xu et al., 2015) non linearity for the predicate classification as described in Eq. 9. Finally, to get the *softmax* distribution of a predicate a fully connected layer (W_{final}) along with the Frequency Baseline (Zellers et al., 2018) has been added to model as described in Eq. 10.

$$rel_{emb} = \text{LReLU}(f_{rel}([n_i^{final}, e_{i,j}^{final}, n_j^{final}])) \tag{9}$$

$$Pr(R|B,O,I) = \text{softmax}(W_{final}(rel_{emb}) + fq(sub, obj)) \tag{10}$$

3. Experiments

3.1. Dataset and Experimental Setup

We used Visual Genome(VG) (Krishna et al., 2017) for our training and evaluation. It is one of the largest and most challenging dataset on scene graph generation for real world images. To have a fair comparison with present state-of-the-art models (Zellers et al., 2018; Newell & Deng, 2017; Zhang et al., 2019b;a), we have used the same refined version of VG proposed in (Xu et al., 2017) along with their official split. This dataset contains the most frequently occurring 150 objects and 50 relationships of VG. We have followed the same evaluation as in the current benchmark (Zhang et al., 2019b) and computed scene graph classification (SGCLS) and predicate classification (PREDCLS).

3.2. Results and Discussion

Table 1, shows the performance of our method in comparison with other methods. Here, methods such as (Xu et al., 2017; Zellers et al., 2018) have used various techniques of context propagation, while ReIDN(Zhang et al., 2019b) and VRU (Zhang et al., 2019a) have used special losses (E.g. contrastive loss) for better modeling of scene graph. Table 1 demonstrates that our novel context propagation for both objects and edges significantly improves the performance even with simple cross-entropy loss.

Additionally, analysis of false prediction provide a great insight that the network learned semantically plausible closer outputs. E.g. 'on' is the most mispredicted relation in evaluation settings, which is 56.9% times predicted as 'of' for (sub., obj.) like (face, woman), (wing, plane). Interestingly, the mispredicted 'face of woman' is more appropriate than 'face on woman', indicating a network not necessarily failing to predict correctly rather due to a huge bias in the dataset to 34.3% 'on' relations. More such positive and negative examples have been listed out in Supplementary. Also, as mentioned in "No Graph Constraint", the high recall in PREDCLS (98.8%) indicates even if the network failed to predict the actual relation in top1, it is mostly captured when multiple relations are being allowed in the subject-object pair.

4. Conclusion

In this paper, we presented a novel approach for scene graph generation by exploiting local and global interaction of other objects. The proposed model is based on a novel customization of the transformer architecture with integrated N2N, E2N, and E2E attention. Additionally, we have generated a visualization of the attention heatmaps to provide insight into the working of the model[3]. Our method improves benchmarks on the Visual Genome dataset.

[3] see the supplementary material

References

Ghosh, S., Burachas, G., Ray, A., and Ziskind, A. Generating natural language explanations for visual question answering using scene graphs and visual attention. *arXiv preprint arXiv:1902.05715*, 2019.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D. A., et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

Li, Y., Ouyang, W., Zhou, B., Wang, K., and Wang, X. Scene graph generation from objects, phrases and region captions. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1261–1270, 2017.

Liu, Y., Wang, R., Shan, S., and Chen, X. Structure inference net: Object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6985–6994, 2018.

Newell, A. and Deng, J. Pixels to graphs by associative embedding. In *Advances in neural information processing systems*, pp. 2171–2180, 2017.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., and Manning, C. D. Generating semantically precise scene graphs from textual descriptions for improved image retrieval. In *Proceedings of the fourth workshop on vision and language*, pp. 70–80, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Xu, B., Wang, N., Chen, T., and Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

Xu, D., Zhu, Y., Choy, C. B., and Fei-Fei, L. Scene graph generation by iterative message passing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Zellers, R., Yatskar, M., Thomson, S., and Choi, Y. Neural motifs: Scene graph parsing with global context. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840, 2018.

Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., and Elhoseiny, M. Large-scale visual relationship understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 9185–9194, 2019a.

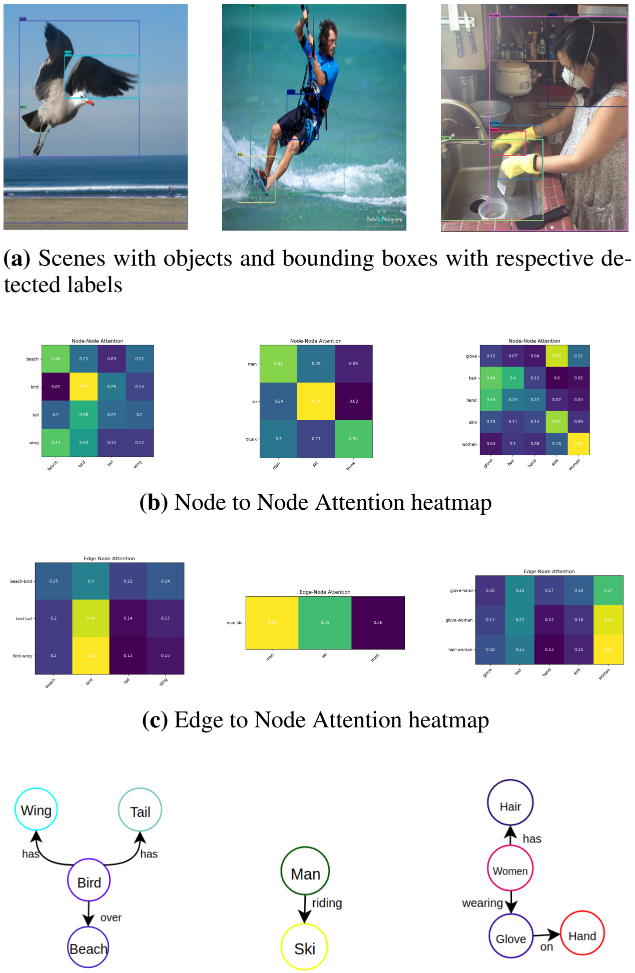
Zhang, J., Shih, K. J., Elgammal, A., Tao, A., and Catanzaro, B. Graphical contrastive losses for scene graph parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 11535–11543, 2019b.

Supplementary Material

This is a supplementary material for our paper. Here, we will discuss more about attention map and qualitative results.

- **Analysis of Attention:** Here, we present an analysis of how attention mechanisms help in scene understanding. In our approach, attention has been used for context propagation between node-node, edge-node as well edge-edge relationships. This interaction has been visualized using an attention heatmap in Fig. 3. Here mutual influence between each pair or row and column is plotted using a score between 0 to 1, where 1 signifies maximum influence, 0 is for minimum. We have used attention mask from top most layer for both module.
  In Fig. 3 (left), a scene with a seagull flying near the beach is shown. Its corresponding node to node (N2N) attention map exhibits detected objects like 'bird', 'wing', 'tail', 'beach' and indicates which nodes or objects are more influential for joint object and relation classification. For example, the node 'bird ' has high attention for 'bird', 'wing', 'tail', that suggests what are the nodes related to it and what could be their potential relationships. Moreover, 'wing' has high attention with 'beach' that could be a potential indicator of influence, suggesting relationship could be flying over the beach. This is further confirmed by attention score for edge 'beach-bird' in edge to node (E2N) attention. For other edges like 'bird-tail ', 'bird-wing', 'bird' could be the most influential node for these edges, thus provide a clear intuition about the kind of relationship that could exists among these nodes.
  In Fig. 3 (middle), nodes like 'man', 'trunk', 'ski' and their mutual high attention score provide context interpretability. Also, its associated edge 'man-ski' shows high influence for all nodes, that reflects context awareness of the edge. Similarly, in Fig. 3 (right), the nodes like 'glove', 'hair', 'hand', shows high mutual influence in node to node(N2N) attention heatmap. Also, 'glove'and 'sink' show high attention indicating contextual influence. The relationships are further derived from edge to node (E2N) attention where edges like 'glove-woman', 'hair-woman', 'glove-hand' show high attention with node 'woman' suggesting that the scene consists of a woman who has hair and that the woman is wearing glove on her hand.

- **Qualitative Results:** In this section, we will provide a few more qualitative samples generated by our network in both positive and negative scenarios. To improve visibility and interpretability, we only consider the interaction among ground truth objects and relations in these examples.



(a) Scenes with objects and bounding boxes with respective detected labels

(b) Node to Node Attention heatmap

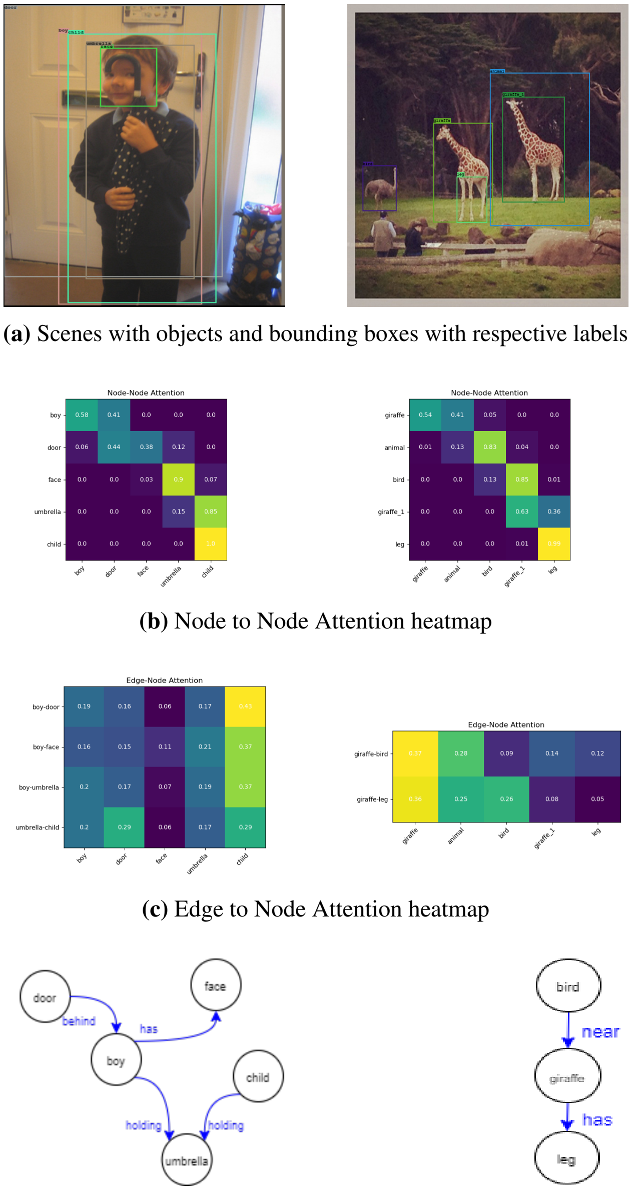
(c) Edge to Node Attention heatmap

(d) Generated scene graph

Figure 3. Some example output from our network with associated attention map and scene graph.

Fig. 4(left column), shows the positive scenario, where our network is able to detect correct relationships label despite the presence of repetitive bounding box (boy and child) or similar objects (giraffe). Thus, it shows the robustness of the method.

Fig. 5(right column), shows the negative scenario, where network prediction is different from ground truth labels. In most of these cases, it was found that predicted labels are semantically closer to ground truth labels, and from a human perspective, both could be right. For example *man-at-beach* and *man-on-beach* both are grammatically correct.



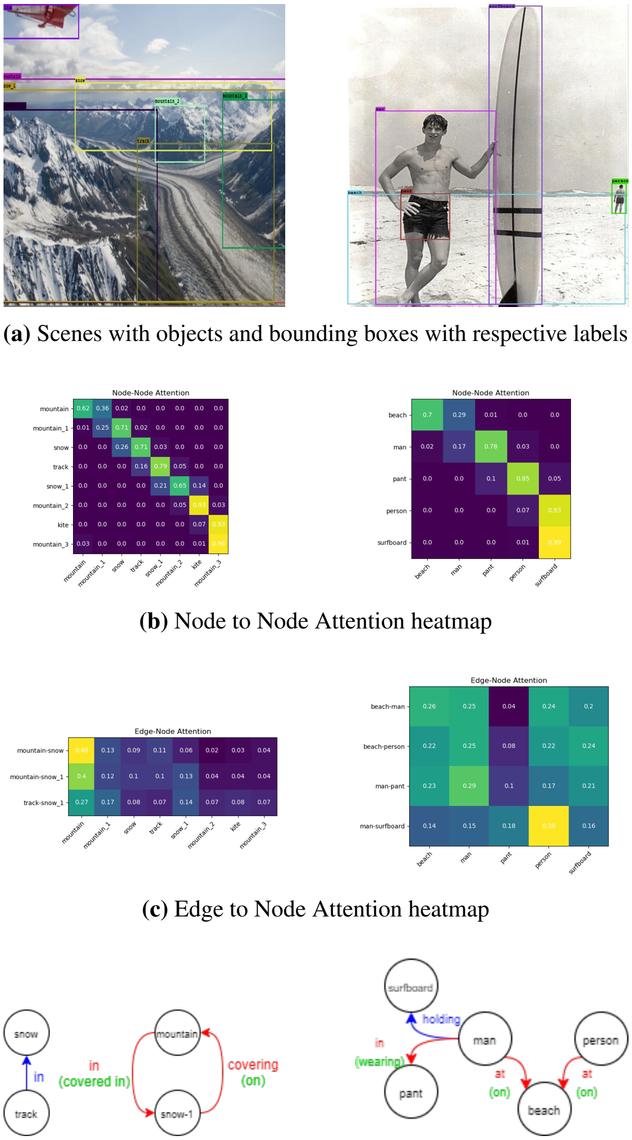
(a) Scenes with objects and bounding boxes with respective labels

(b) Node to Node Attention heatmap

(c) Edge to Node Attention heatmap

(d) Generated scene graphs

Figure 4. Some positive example outputs from our network with associated attention map and scene graph.



(a) Scenes with objects and bounding boxes with respective labels

(b) Node to Node Attention heatmap

(c) Edge to Node Attention heatmap

(d) Generated scene graph. Here, blue ones are correctly predicted, red ones are mispredicted and green ones are the correct ground truth label for each mispredicted label.

Figure 5. Some negative example outputs from our network with associated attention map and scene graph.