Relation Transformer Network

Rajat Koner[1], Poulami Sinhamahapatra[2], and Volker Tresp[1,3]

[1] Ludwig Maximilian University, Munich, Germany
[2] Fraunhofer IKS, Munich, Germany
[3] Siemens AG, Munich, Germany
koner@dbs.ifi.lmu.de, poulami.sinhamahapatra@iks.fraunhofer.de,
volker.tresp@siemens.com

Abstract. The identification of objects in an image, together with their mutual relationships, can lead to a deep understanding of image content. Despite all the recent advances in deep learning, in particular the detection and labeling of visual object relationships remain a challenging task. In this work, we present the Relation Transformer Network, which is a customized transformer-based architecture that models complex object to object and edge to object interactions, by taking into account global context. Our hierarchical multi-head attention based approach efficiently models and predicts dependencies between objects and their contextual relationships. In comparison to other state of the art approaches, we achieve an absolute mean 3.72% improvement in performance on the Visual Genome dataset.

Keywords: Scene Graph, Scene Understanding, Transformer

1 Introduction

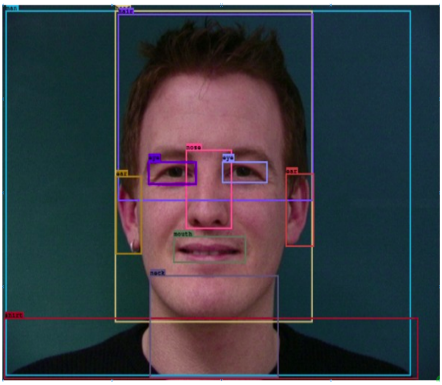
A *scene graph* is a graphical representation of an image consisting of multiple entities and their interacting relationships expressed in triplet format like $\langle subject, predicate, object \rangle$. Objects in the scene can be represented as *nodes* in a graph, and their mutual relationships as directed *edges* in the graph, labeled by the predicate.[4] For example, in Figure 1, ‘Eye’,‘Hair’,‘Head’,‘Man’ are objects or nodes, and their mutual relationships are described by the predicates ‘has’,‘on’.

Scene graph generation is executed in two steps: First, the detection of the objects present in the image and, second, the identification of the predicates. Current state of the art object detection approaches have achieved very good performance in spatially locating various objects in an image. On the contrary, current state of the art models for relation prediction are still in a nascent stage. To achieve state-of-the art performance, it is important to consider complex context information and to model the dependencies between objects and predicates.

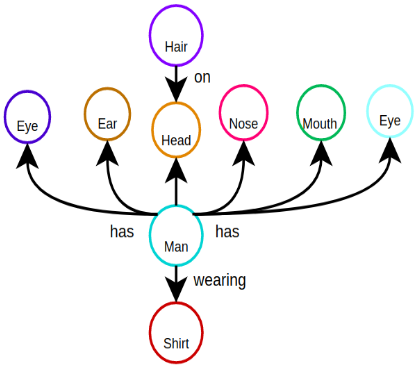
[4] In this paper, objects or nodes, generic relationship between two objects or edge, directed relationship between objects or predicates is used interchangeably.

Am extracted scene graph can be used in many applications like visual question answering[5], image retrieval[28], image captioning[15].

In this paper, we propose a novel scene graph generation algorithm, called **Relation Transformer Network**, which leverages interactions among objects, predicates, their respective influence on each other and their co-occurrence patterns. The inspiration to build this network came from our investigation on current work on scene graph generation. In the following section, we will describe our approach with a focus on two important challenges that need to be addressed.



(a) Scene with face of a man



(b) Corresponding scene graph

Fig. 1: 1a is an example image with 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.

2 Challenges in Scene Graph Detection

Firstly, in an image it is important to understand the role of each object and how objects are related and influenced by others in the context of the whole image. For example, in Figure 1, the presence of nodes like 'Eye ', 'Hair', 'Nose', 'Head', indicates that these describe a face and relationships between face segments. Also, presence of the node 'Shirt' implies that this is a face of 'Human'. Node dependencies are also important for predicting an edge or a pairwise relation. Conversely, spatial and semantic co-occurrence also help in identifying node classes. We have modeled this Node to Node (N2N) dependency using a novel transformer [31] encoder based architecture. Figure 2(b) shows our network architecture for multi-hop node context information propagation. Although, some recent works [36,40,7,37] have used various other methods (like Bidirectional LSTM[8], GCN[12]) to propagate context for each node. We propose that our transformer-based multi-hop N2N context propagation approach enhances information flow and gives better performance.

Secondly, the most important challenge is to predict correct predicates, describing the relationship among two objects. Here, we have applied two novel techniques: 1) Edge to Node (E2N) attention which takes into account each node's influence on an edge, and 2) Edge to Edge (E2E) attention, which appraises the impact of other edges on an edge. The E2N attention helps to exploit semantic and spatial co-relation among various objects. Consider Figure 1. If the edge between 'Man' and 'Shirt' is aware of other neighboring nodes like 'Hair', 'Face', 'Head' then E2N attention could give a more confident prediction on 'wearing' compared to other predicates. Our E2E attention exploits this interaction even further by accumulating context from other edges (composed with similar objects). For example, in Figure 1, when we know the relation between 'Man' and 'Ear' is 'Has' then the relation between 'Man' and other closely related node of 'Ear '(like 'Eye', 'Nose', even other 'Ear') will be the same or close to it. Figure 2 (c) depicts the network architecture for E2N and E2E attention propagation based on our transformer decoder. The attention map of both E2N and E2E can be used to visualize the influence. Additionally, the embedding vector we get from nodes and edges could be useful for further tasks like graph embedding.

We have tested our approach on the Visual Genome[13] dataset, and achieved better than state of the art results. In particular, we have gained an absolute margin of maximum 7.3% improvement in scene graph classification and maximum 1.2% improvement in predicate classification compare to the current state of the art[43]. Also, we have conducted experiments to empirically verify our concept. To summarize our contribution :

- We propose a novel customized transformer-based architecture for object context propagation using N2N, E2N and E2E attention to exploit the various interactions and influences of all nodes and edges for scene understanding.
- Our novel node and edge context enrichment module can be used to generate high-quality cross-modal relational embeddings that could be used for other vision-language tasks.
- Our Relation Transformer Network gained a mean absolute 3.72% improvement and has achieved new benchmark results on Visual Genome. Extensive ablation studies and the analysis of attention maps provide an inside view of the working of the network.

3 Proposed Model/Methods

In this section, we present problem definition and describe the proposed Relation Transformer Network. An overview of the network architecture is shown in Figure 2.

[4] All N2N,E2N,E2E and POS FFN block followed by [Add & Norm] block and a residual connection like Transformer.

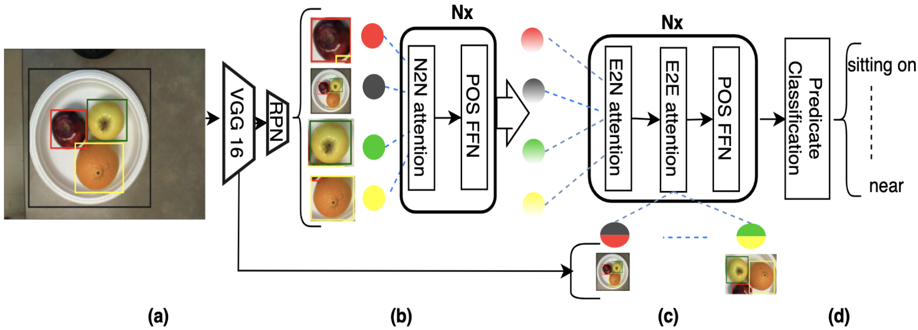


Fig. 2: An overview of the proposed Relation Transformer Network. The model decomposes the task in four-stages: a)features generation by an object detector and bounding box extraction, b)creation of node embedding by accumulating context from every node, c)creation of edge embedding by accumulating scene context from all nodes and then from other edges, d)classification of the relation using $\langle subject, edge, objects\rangle$ manner. All nodes are represented in dark color box (left), context rich nodes are in light color (middle) and edges are in two color (below) based on the creation node. Blue dotted line describes the attention. Best viewed in color.

3.1 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 "semantic", "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 $r_{s\rightarrow o} \in R$ defines the relationship between the subject and object node. So, scene graph generation can be formulated as a three factor model as,

$$Pr(G|I) = Pr(B|I)\; Pr(O|B, I)\; Pr(R|O, B, I). \tag{1}$$

$Pr(B|I)$ can be inferred by any object detection model. We have used Faster RCNN[27] for this task (Sec. 3.2). Determining the conditional probability of a object class $Pr(O|B, I)$, where the presence of one object can be influenced by the presence of another, is handled in Sec. 3.3.2. To model the relationships $Pr(R|O, B, I)$ among objects, we first compute an undirected edge (Sec. 3.3.3) among two objects, and then conclude on a directed edge (Sec. 3.4) or relation($r_{s\rightarrow o}$) from subject to object.

3.2 Object Detection

We have used Faster RCNN[27] with a VGG-16[30] backbone for object detection. For each N object proposal, we get initial visual features $v_i^{RoI} \in \mathbb{R}^{4096}$, bounding box $b_i \in \mathbb{R}^5$[5] and object class probability $o_i^{init} \in \mathbb{R}^{200}$. We considered these individual proposals as initial nodes of the scene graph. Their concatenated features $n_i^{in} \in \mathbb{R}^{2048}$ are initial node features. In order to contextualize each node by the next module (Sec. 3.3.2), and to reduce the dimensionality, a linear projection(f_{nlp}) layer has been used, as

$$n_i^{in} = f_{nlp}([v_i^{RoI}, o_i^{init}, b_i]), where, \;\; i = 1..n \tag{2}$$

3.3 Context Propagation:

The core of our network is based on context propagation across all nodes and edges. It uses a customized transformer encoder-decoder architecture[31] for nodes and edges. At the heart of the transformer is a self-attention mechanism. In the subsequent section we briefly review some important components of the attention approach.

3.3.1 Attention: Attention mechanisms allow modeling of dependencies without regard to their distance in the input sequence and allow efficient information propagation compared to Recurrent Neural Networks (RNN). Attention is being largely used in language and vision-related tasks in order to map contextual information. The transformer[31] architecture uses self-attention mechanisms for drawing global dependencies instead of sequence aligned RNNs. One defines

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

The last equation describes a self-attention function, where query(Q), keys(K), 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 with a query matrix and its corresponding key. In Sec. 3.3.2, we have used object-to-object or node-to-node (N2N) or encoder self-attention, and in Sec. 3.3.3 we have used E2N or decoder-encoder self-attention, and E2E or decoder-decoder self attention.

3.3.2 Context Propagation for Objects: Contextualization of objects not only enhances object detection [16] by exploring the surroundings of objects, but it also encodes more expressive features for relation classification. For N discrete nodes, we have used initial features from Eq. 2 along with a positional ($PE^n \in \mathbb{R}^{2048}$) feature vectors, based on the actual position of the nodes [31] in the sequence. In particular,

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

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

[5] check the Sec. 4.2.1 for more details

After contextualization of these N nodes by Eq. 4, we got final node features (n_i^{final}). In the next steps these node features are used for two purposes. First, they are passed to a linear object classifier (Eq. 5) to get the final object class ($o_i^{final} \in C$) probability and, second, these nodes features are passed to the next module for edge context propagation.

3.3.3 Context Propagation for Edges Perhaps the most important part in scene graph generation or detection depends on the expressiveness of the edge features and how well they can depict relations among object pairs. In this module, we try to capture edge features by accumulating context information across all nodes and edges. An edge is highly dependent on the local context, as it comes from only two source nodes (subject, object). This inspires many current scene graph generation models [15,34,32,43,6] which process the particular subject, object and edge between them separately. We have created the initial edge embedding ($e_{i,j}^{in} \in \mathbb{R}^{2048}$) from the two objects, and then allowed the network to learn the influences from other nodes or edges to exploit both local and global contexts.

Similar to a node embedding, an edge consists of combined visual, spatial and semantics features of two possible object combinations. Here, visual features $e_{ij}^{vis} \in \mathbb{R}^{4096}$ come from the ROI feature map of union of two object boxes $b_{i,j}$ passed trough the VGG top layer as shown in Figure 2. Afterwards, the edge-specific binary-mask spatial feature (e_{ij}^{spt}) is combined with visual features[6]. Also, spatial features $b_{i,j} \in \mathbb{R}^5(b_i$ and $b_j)$ have been added. We use concatenated GLOVE vector[23] embedding of the class of two objects. (o_i^{final} and o_j^{final}) are the semantic embedding (e_{ij}^{sem}) for the edges. We get an initial edge features from Eq. 6, where f_{elp} is a linear projection layer as a transformer accepts the same size input for both encoder and decoder. We get,

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

Our edge context enrichment module is a modified transformer decoder[31]. There are three main modifications we have incorporated in our network such that we can use it for edge context propagation. At first, we removed the decoder masked attention so that it can attend the whole sequence, not only part of it. Secondly, our positional encoding vector ($PE_{edge_{ij}}$) for edge ($e^{in_{i,j}}$) encodes the position of both the source nodes, instead of the actual position of the edge. Our hypothesis is that it will be helpful for the network to distinguish the source nodes (subject and object) out of all N distinct nodes and an edge to other edges by keeping track of their source. This design bias can accumulate a global context without losing its focus on local context or source nodes. We get,

$$\begin{aligned} PE_{(k,k+1)}^{e_{ij}} &= [\sin(pos_i/100^{2k/d_{dim}}), \cos(pos_i/100^{2k/d_{dim}})]. \\ PE_{(k+2,k+3)}^{e_{ij}} &= [\sin(pos_j/100^{2k/d_{dim}}), \cos(pos_j/100^{2k/d_{dim}})]. \end{aligned} \tag{7}$$

[6] check the Sec. 4.2.1 for more details.

Eq. 7 describes positional encoding for an edge, where pos_i and pos_j are the positions of the nodes n_i and n_j, $d_{dim} \in \mathbb{R}^{2048}$ is same dimension as $e^{in}_{i,j}$ and k denotes the k^{th} position in the features vector.

Third, we have changed the order of self-attention applied in the decoder. First we have applied E2N (decoder-encoder) self-attention from an edge to all the nodes, then E2E (decoder-decoder) attention from an edge to all the edges. As the edge is created from only two nodes, so E2N attention provide necessary global context from all nodes. Afterwards, for an edge enriched with global context, E2E attention should help to learn from edges which have similar semantics or embedding, as the relation they encode could also be the similar. Finally, we get contextual edge features ($e^{final}_{i,j} \in \mathbb{R}^{2048}$) from Eq. 8, as

$$e^{final}_{i,j} = \text{decoder}(e^{in}_{i,j} + PE^{e_{ij}}). \tag{8}$$

3.4 Relation Classification

As a relation is directional, it can change if the order of *subject* or *object* is changed. Thus, after getting the context rich node and edge embedding, we create a joint relational embedding ($Rel_{emb} \in \mathbb{R}^{2048}$) consisting of triples like $\langle subject, edge, object \rangle$ for the predicate classification described in Eq. 9. Afterwards we applied Leaky ReLU[35] nonlinearity followed by *softmax* distribution of predicates. Frequency Baseline[40] is also been added to network prediction.

$$Rel_{emb} = f_{rel}([n^{final}_i, e^{final}_{i,j}, n^{final}_j]). \tag{9}$$

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

Finally, we obtain the relation among two objects as described in Eq. 10.

4 Implementation and Experimental Details

In this section, we will describe the dataset, and explain implementation details of our network pipeline and spatial embedding implementation.[7]

4.1 Datasets

We used one of the large scene graph dataset, i.e., Visual Genome[13], for our experimental evaluation. It is one of the most challenging and current stateof art dataset for scene graph detection and generation for the real world images. The original dataset consists of 108,077 images with annotated object bounding boxes, class and interlinked relation among the objects. These annotations are

[7] soon we will release our code with trained weight.

quite noisy: e.g., multiple bounding boxes are provided for a single object. To alleviate this problem Lu at el. [36] proposed a refined version of the dataset, which consists of the most frequently occurring 150 objects and 50 relationships. To have a fair comparison with most of the present state of art model[40,21,43,7,42] we have used this refined dataset. Also, our train (55K), validation (5K) and test (26K) split is same as per the dataset.

4.2 Implementation Details

We have implemented our model in pythorch-0.4[22] and trained the model in a single Nvidia GTX 1080 ti GPU. The input of our model is the same as [40], that is an image with a size of 592 × 592. To have a fair comparison, we have used Faster-RCNN[27] with vgg16[30] backbone pretrained on visual genome dataset as per [40,43]. As mentioned in our Sec. 3.3.2 and Sec. 3.3.3, the encoder and the decoder module accept input features of size 2048. We have used 3x encoder, 2x decoder and 12 attention head for transformer network. The glove vector embedding has size 200. SGD with momentum along with learning rate of 10^{-3} and batch size of 6 has been used. Also, we have used cross-entropy loss for both of our object and relation classification loss.[8] In training, we used 1 positive edge for 4 negative edges, and randomly flip some images as part of data augmentation. We have followed the same evaluation as in current benchmark[43] and computed scene graph classification (SGCLS) and predicate classification (PREDCLS).

4.2.1 Spatial Embedding We postulate that relationships among two objects also depend on their spatial location. For example in the Figure 2 the relation (sitting on) among apple (in red box) and plate (in black box) can be inferred through spatial location. We have encoded spatial information using the normalize position[27] of nodes and edges, and spatial masks[40,3] of subject and object nodes. A normalized coordinate features of b_i and union of two bounding boxes(b_{ij}) can be expressed as,

$$b_{\text{norm}} = (\frac{x}{w_{img}}, \frac{y}{h_{img}}, \frac{x+w}{w_{img}}, \frac{y+h}{h_{img}}, \frac{wh}{w_{img}h_{img}}) \tag{11}$$

where bounding boxes are provided in the format (x, y, w, h), and w_{img},h_{img} are the width and height of the image. To leverage more on spatial embedding, we have used a binary mask of two boxes b_i and b_j and fed them to a conv layer specified in [40]. Afterwards this spatial features was added with edge visual features (e_{ij}^{vis}).

5 Results

In this section, we will describe our results and analyze the interpretability of output using attention heatmap.

[8] List of all hyper-parameters are given in the supplementary material.

Model	**SGCLS**			**PRDCLS**		
Recall@	20	50	100	20	50	100
VRD[17]	-	11.8	14.1	-	27.9	35.0
Message Passing[36]	31.7	34.6	35.4	52.7	59.3	61.3
Associative Embedding[21]	18.2	21.8	22.6	47.9	54.1	55.4
MotifNet(Left to Right)[40]	32.9	35.8	36.5	58.5	65.2	67.1
Permutation Invariant[7]	-	36.5	38.8	-	65.1	66.9
Large Scale VRU[42]	36.0	36.7	36.7	66.8	68.4	68.4
ReIDN[43]	36.1	36.8	36.8	66.9	68.4	68.4
Relation Transformer (Ours)	**43.4**	**43.6**	**43.7**	**68.1**	**68.5**	**68.5**

Table 1: Comparison of our model with state of the art methods tested in Visual Genome[13]

Table 1, shows the performance of our Relation Transformer Network in comparison with other methods. Here, methods like Message Passing[36], MotifNet[40] and Permutation Invariant[7] have also used context to model relationships. It demonstrates that our novel attention based context propagation for both object and edge significantly improves relationship detection.

5.1 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.

6 Ablation Study

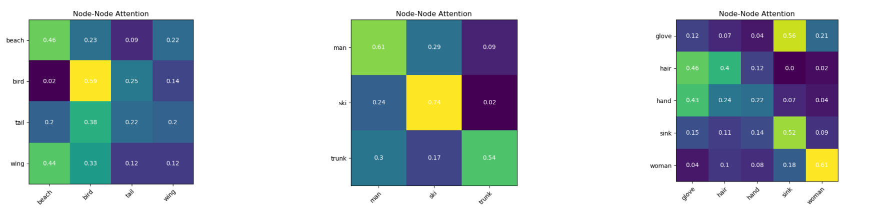
To show the efficacy of our proposed novel object and edge context enrichment modules, we have conducted several ablation experiments. Object and edge context propagation module are both able to perform multi-hop attention propagation between objects and edges. Table 2 compares the empirical performance of these two modules for various hops. From these results, we can clearly infer that both modules have significantly contributed to relationship classification. In this study, some interesting observations are summarized below:

- All nodes need to be contextualized well enough, in order to propagate context for edges. This is demonstrated in Table. 2, as adding more edge context layer without adequate object context propagation module harms performance. Thus, it shows the importance of contextualized objects.
- After objects are properly contextualized, adding more edge context modules on top of object context modules increases the performance. Jointly it shows the importance of both the modules.
- If we use large the number of layers (three) for both modules then performance decreases. One of the possible reason, as the network size grows it becomes hard to optimize.
- We have also conducted experiments without using any E2E attention in our final configuration (3 object context, 2 edge context layer), and the result did not change significantly. We assume N2N and E2N provides sufficient contextualization. This remains an open question and requires further analysis.

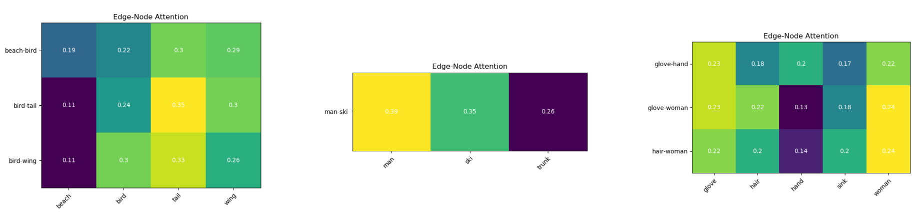
7 Related Work

Scene Understandings with Language Priors: Scene understating evolves through many phases throughout the past decade. Initially researcher tried to localize objects or regions in a image-based on given caption or text reference[19,20,9,24] to understand the scene. These approaches mostly match the referenced language to the part of the image, but their lack of expression and graph structure hinders true understanding of an image. Later Johnson et. al[11] introduced scene graph, and Lu et. al[17], proposed the visual relationship detection with language

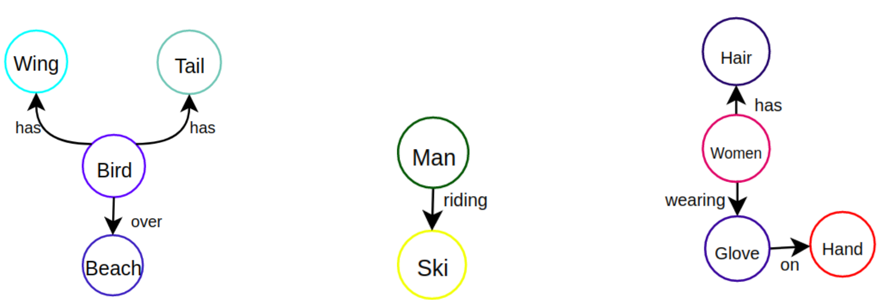
(a) Scenes with objects and bounding boxes with respective detected labels



(b) Node to Node Attention heatmap



(c) Edge to Node Attention heatmap



(d) Generated scene graph

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

Object Context	Edge Context	Predicate Classification R@20	R@50	R@100
1	1	67.2	67.5	67.6
2	1	67.7	68.1	68.1
1	2	67.6	68.0	68.0
2	2	67.8	68.2	68.2
3(ours)	**2(ours)**	**68.1**	**68.5**	**68.5**
3	3	67.9	68.3	68.3

Table 2: Impact of Object context module and Edge context module

priors with its associated dataset named VRD. Scene graph of an image gives a semantically interpretable graph structure. Some of the early research also [2,39] used language priors with a knowledge graph or external knowledge base to improve the relation prediction. Several works have tried to address the problem by combining the visual and other semantic features of the subject, object and predicates individually[43,15,21,32,38,41,17] by enriching the features with a new type of loss, pooling, and representation or using other modalities[29].

Context in Scene Graph: Contextual information proven to be helpful for object detection[16], visual question answering[1], scene understandings[20]. Recent advancement using attention in NLP[31], convolution network[33] provided an efficient way to model complex interaction of entities. Also, the evolution of the graph structure network especially graph convolution[12] helps to propagate context and produced state of the art result. Inspired from these advancement various recent relationship detection network tries to incorporate local or global context with [34,37] or without attention [26,40,7,36]. Context helps to predict the mutual co-occurrence of object presence and better modeling of the global scene understandings. Our work is also used attention based context propagation, and closely related to Neural Motif[40]. In the paper [40], authors use context propagation for objects, that enriches each object with more global semantics features, that helps to classify the relations. Our works differ significantly in that we not only consider the possibility of the mutual co-occurrence objects, also how the presence of objects or predicates jointly influences each other. This novel approach enables more information exchange across all objects and predicates at different stages of the network and dynamically learns their influence using attention.

Transformer in Vision: After the release of transformer[31] followed by BERT[4], it became one of the most popular choices for various pre-training cross-modal task. In the Vision-Language pretraining task, BERT[4] style architecture becomes a default choice for its ability to process non-sequential data and in almost all cases it produced a new state of the art results. In [18], a two-stream network for joint vision-language modalities has been used to get an enhanced representation for a task like visual question answering, image captioning. Some contemporary work, like [14] uses a combination of sentences and image patches jointly for pretraining and achieved state of the result on GQA[10] or task like Masked Object Classification(MOC), Visual Linguistic Matching(VLM). A very recent work[25] uses BERT for large scale joint object embedding. This recent surge shows the importance and efficacy of the Transformer and BERT style of architecture. Although, instead of pretraining, we have used the transformer network with marginal modifications that it could focus more on how a object and predicate interact and able to capture both local and global context.

8 Conclusion

In this paper, we presented an approach for complex visual scene analysis using scene graphs by exploiting local and global influences. The proposed models is based on a novel customized transformer based architecture, coined as *Relation Transformer Network* with integrated N2N, E2N and E2E attention. Additionally, we have generated a visualization of the attention heatmaps to provide insight in the working of the model. Our method improves on Visual Genome dataset benchmarks. Our node and edge context enrichment modules can generate a cross-modal relational graph embedding of an image. Future research will focus on how relational embedding and context propagation could impact various vision and language tasks, as well as in scene understanding.

References

1. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and visual question answering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018) 12
2. Baier, S., Ma, Y., Tresp, V.: Improving visual relationship detection using semantic modeling of scene descriptions. In: International Semantic Web Conference. pp. 53–68. Springer (2017) 12
3. Dai, B., Zhang, Y., Lin, D.: Detecting visual relationships with deep relational networks. In: Proceedings of the IEEE conference on computer vision and Pattern recognition. pp. 3076–3086 (2017) 8
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 12
5. Ghosh, S., Burachas, G., Ray, A., Ziskind, A.: Generating natural language explanations for visual question answering using scene graphs and visual attention. arXiv preprint arXiv:1902.05715 (2019) 2
6. Gu, J., Zhao, H., Lin, Z., Li, S., Cai, J., Ling, M.: Scene graph generation with external knowledge and image reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1969–1978 (2019) 6
7. Herzig, R., Raboh, M., Chechik, G., Berant, J., Globerson, A.: Mapping images to scene graphs with permutation-invariant structured prediction. In: Advances in Neural Information Processing Systems. pp. 7211–7221 (2018) 2, 8, 9, 12
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997) 2
9. Hu, R., Rohrbach, M., Andreas, J., Darrell, T., Saenko, K.: Modeling relationships in referential expressions with compositional modular networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1115–1124 (2017) 10
10. Hudson, D.A., Manning, C.D.: Gqa: A new dataset for real-world visual reasoning and compositional question answering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6700–6709 (2019) 13
11. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3668–3678 (2015) 10
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016) 2, 12
13. 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) 3, 7, 9
14. Li, G., Duan, N., Fang, Y., Jiang, D., Zhou, M.: Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. arXiv preprint arXiv:1908.06066 (2019) 12
15. Li, Y., Ouyang, W., Zhou, B., Wang, K., 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) 2, 6, 12
16. Liu, Y., Wang, R., Shan, S., 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) 5, 12
17. Lu, C., Krishna, R., Bernstein, M., Fei-Fei, L.: Visual relationship detection with language priors. In: European conference on computer vision. pp. 852–869. Springer (2016) 9, 10, 12
18. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Advances in Neural Information Processing Systems. pp. 13–23 (2019) 12
19. Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., Murphy, K.: Generation and comprehension of unambiguous object descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 11–20 (2016) 10
20. Nagaraja, V.K., Morariu, V.I., Davis, L.S.: Modeling context between objects for referring expression understanding. In: European Conference on Computer Vision. pp. 792–807. Springer (2016) 10, 12
21. Newell, A., Deng, J.: Pixels to graphs by associative embedding. In: Advances in neural information processing systems. pp. 2171–2180 (2017) 8, 9, 12
22. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035 (2019) 8
23. Pennington, J., Socher, R., 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) 6
24. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: Proceedings of the IEEE international conference on computer vision. pp. 2641–2649 (2015) 10
25. Qi, D., Su, L., Song, J., Cui, E., Bharti, T., Sacheti, A.: Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. arXiv preprint arXiv:2001.07966 (2020) 13
26. Qi, M., Li, W., Yang, Z., Wang, Y., Luo, J.: Attentive relational networks for mapping images to scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3957–3966 (2019) 12
27. Ren, S., He, K., Girshick, R., 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) 4, 8
28. Schuster, S., Krishna, R., Chang, A., Fei-Fei, L., 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) 2
29. Sharifzadeh, S., Berrendorf, M., Tresp, V.: Improving visual relation detection using depth maps. arXiv preprint arXiv:1905.00966 (2019) 12
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 4, 8
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017) 2, 5, 6, 12
32. Wan, H., Luo, Y., Peng, B., Zheng, W.S.: Representation learning for scene graph completion via jointly structural and visual embedding. In: IJCAI. pp. 949–956 (2018) 6, 12

33. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7794–7803 (2018) 12
34. Woo, S., Kim, D., Cho, D., Kweon, I.S.: Linknet: Relational embedding for scene graph. In: Advances in Neural Information Processing Systems. pp. 560–570 (2018) 6, 12
35. Xu, B., Wang, N., Chen, T., Li, M.: Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853 (2015) 7
36. Xu, D., Zhu, Y., Choy, C.B., Fei-Fei, L.: Scene graph generation by iterative message passing. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017) 2, 8, 9, 12
37. Yang, J., Lu, J., Lee, S., Batra, D., Parikh, D.: Graph r-cnn for scene graph generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 670–685 (2018) 2, 12
38. Yin, G., Sheng, L., Liu, B., Yu, N., Wang, X., Shao, J., Change Loy, C.: Zoom-net: Mining deep feature interactions for visual relationship recognition. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 322–338 (2018) 12
39. Yu, R., Li, A., Morariu, V.I., Davis, L.S.: Visual relationship detection with internal and external linguistic knowledge distillation. In: Proceedings of the IEEE international conference on computer vision. pp. 1974–1982 (2017) 12
40. Zellers, R., Yatskar, M., Thomson, S., 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) 2, 7, 8, 9, 12
41. Zhang, H., Kyaw, Z., Chang, S.F., Chua, T.S.: Visual translation embedding network for visual relation detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5532–5540 (2017) 12
42. Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., Elhoseiny, M.: Large-scale visual relationship understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 9185–9194 (2019) 8, 9
43. Zhang, J., Shih, K.J., Elgammal, A., Tao, A., Catanzaro, B.: Graphical contrastive losses for scene graph parsing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11535–11543 (2019) 3, 6, 8, 9, 12