Scenes and Surroundings: Scene Graph Generation using Relation Transformer

Anonymous Authors

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 \(<subject, predicate, object>\). 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 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

\[O \rightarrow N \rightarrow E\]

\[E \rightarrow \text{Node}_i \rightarrow E\]

\[E \rightarrow \text{Edge}_i \rightarrow E\]

\[\text{Node}_i \rightarrow \text{Edge}_i \rightarrow \text{Node}_j\]

\[\text{Node}_i \rightarrow \text{Edge}_i \rightarrow \text{Node}_j \rightarrow \text{Edge}_i \rightarrow \text{Node}_k\]

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.
Scene Graph generation using Relation Transformer

1. Scene consisting of a man’s face

2. 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.

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 $r_{sub \rightarrow obj} \in R$ defines the relationship between the subject and object node. Hence, scene graph generation can be formulated as a three factor model as,

$$Pr(G[I]) = Pr(B[I]) Pr(O[B, I]) Pr(R[O, B, I])$$

$$Pr(B[I])$$ 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 ($r_{sub \rightarrow obj}$) (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 $o_i^{init} \in \mathbb{R}^{200}$. 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.

$$n_i^{in} = f_{nlp}([v_{i}^{RoI}, o_i^{init}, b_i])$$

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:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

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 ($pos_{enc} \in \mathbb{R}^{2048}$)
After contextualization of the nodes by the encoder\(^2\) in Eq. 4, we have obtained final node features (\(\mathbf{n}_{i}^{\text{final}}\)).

\[
\mathbf{n}_{i}^{\text{final}} = \text{encoder}(\mathbf{n}_{i}^{\text{in}} + \text{pos}\_\text{enc}^{n}(\mathbf{n}_{i}^{\text{in}})).
\]

(4)

And

\[
\alpha_{i}^{\text{final}} = \text{argmax}(f_{\text{classifier}}(\mathbf{n}_{i}^{\text{final}})).
\]

(5)

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 (\(\text{pos}\_\text{enc}^{e_{ij}} \in \mathbb{R}^{2048}\)) for edges (\(e_{ij}^{\text{in}}\)) 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.

\[
\text{pos}\_\text{enc}^{e_{ij}}(k,k+1) = [\sin(p_{i}/m^{2k/d_{\text{dim}}}), \cos(p_{i}/m^{2k/d_{\text{dim}}})].
\]

\[
\text{pos}\_\text{enc}^{e_{ij}}(k+2,k+3) = [\sin(p_{j}/m^{2k/d_{\text{dim}}}), \cos(p_{j}/m^{2k/d_{\text{dim}}})].
\]

(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, \(d_{\text{dim}} \in \mathbb{R}^{2048}\) is same dimension as \(e_{ij}^{\text{in}}\), 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
Scene Graph generation using Relation Transformer

<table>
<thead>
<tr>
<th>Model</th>
<th>Graph constraint</th>
<th>No graph constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SGCLS</td>
<td>PRDCLS</td>
</tr>
<tr>
<td></td>
<td>Recall@20,50,100</td>
<td>Recall@50,100</td>
</tr>
<tr>
<td>Message Passing (Xu et al., 2017)</td>
<td>31.7</td>
<td>34.6</td>
</tr>
<tr>
<td>Associative Embedding (Xu et al., 2017)</td>
<td>18.2</td>
<td>21.8</td>
</tr>
<tr>
<td>MotifNet(Left to Right) (Zellers et al., 2018)</td>
<td>32.9</td>
<td>35.8</td>
</tr>
<tr>
<td>Large Scale VRU (Zhang et al., 2019a)</td>
<td>36.0</td>
<td>36.7</td>
</tr>
<tr>
<td>RelDN (Zhang et al., 2019b)</td>
<td>36.1</td>
<td>36.8</td>
</tr>
<tr>
<td>Relation Transformer (Ours)</td>
<td>43.4</td>
<td>43.6</td>
</tr>
</tbody>
</table>

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

2.4. Relation Classification

A relation is a directional property, i.e., subject and object cannot be exchanged. After obtaining the context-rich node and edge embeddings, a joint relational embedding (rel\_emb \in \mathbb{R}^{2048}) has been created consisting of triplets like (subject, edge, object) 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.

\[
\text{rel}\_\text{emb} = \text{LReLU}(f_{rel}(n_{i,j}^{\text{final}}, e_{i,j}^{\text{final}}, n_{j}^{\text{final}}))
\]  

Eq. 9

\[
Pr(R|B,O,I) = \text{softmax}(W_{final}(\text{rel}\_\text{emb}) + f_{q}(\text{sub}, \text{obj}))
\]  

Eq. 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 (PRDCLS).

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 PRDCLS (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


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.

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

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