Graph-LSTM with Global Attribute for Scene Graph Generation

Shao, Tong; Wu, Dapeng Oliver

Published in:
Journal of Physics: Conference Series

Published: 01/01/2021

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

License:
CC BY

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1088/1742-6596/2003/1/012001

Publication details:
Shao, T., & Wu, D. O. (2021). Graph-LSTM with Global Attribute for Scene Graph Generation. *Journal of Physics: Conference Series*, *2003*, [012001]. https://doi.org/10.1088/1742-6596/2003/1/012001

Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Graph-LSTM with Global Attribute for Scene Graph Generation

To cite this article: Tong Shao and Dapeng Oliver Wu 2021 *J. Phys.: Conf. Ser.* **2003** 012001

Graph-LSTM with Global Attribute for Scene Graph Generation

Tong Shao[1], **Dapeng Oliver Wu**[1]

[1]Department of Electrical and Computer Engineering, University of Florida, 1064 Center Drive, Gainesville FL 32611, USA

E-mail: stlm1991@ufl.edu

Abstract. Lots of machine learning tasks require dealing with graph data, and among them, scene graph generation is a challenging one that calls for graph neural networks' potential ability. In this paper, we present a definition of graph neural network (GNN) consists of node, edge and global attribute, as well as their corresponding update and aggregate functions. Based on this, we then propose a realization of GNN model called Graph-LSTM and use it in scene graph generation. The model first extracts the item features in the image as the initial states of the node-LSTM representing subject/object and edge-LSTM representing predicate. Two LSTMs update the states via LSTM's timestep and aggregate information via message passing. Repeat the update-aggregate until convergence. Meanwhile, the tag feature, i.e., the generated probability distribution of image's semantic concepts is sent to the LSTM through a semantic compositional network (SCN). The SCN-LSTM is trained in an ensemble style, and hence allows the tag feature to serve as the global attribute providing context information to all individuals. The LSTMs' final states are input to inference modules and generate the triplet $(subject, predicate, object)$ of the scene graph. Experimental results show that Graph-LSTM outperforms the Message Passing and the attention Graph Covolutional Network methods, proving the effectiveness of the proposed scheme.

1. Introduction

The rapid development of machine learning, especially deep learning [1], has greatly promoted the development of many applications. For visual information processing, the convolutional neural networks (CNNs) [2] models, e.g. ResNet [3], have achieved great results in object detection [4]. However, those existing methods are object-centric, focusing on detecting single object and search through every pair to find higher-level relations. This high complexity nature makes it hard to work well on graph data, such as the scene graph generation (SGG), which aims at finding the triplet relations $(subject, predicate, obeject)$ in a given image.

Recently, how to process graph data using deep learning methods has become increasingly important, which may significantly improve the SGG. Graphs are a data structure consisting of items (nodes) and their relationships (edges). As traditional deep learning methods are unable to process graph-structured data as input, many graph neural networks (GNNs) have been proposed, including the Graph Convolutional Neural Network (GCN) [5], Message-passing neural network (MPNN) [6] and non-local neural network (NLNN) [7]. However, current GNN models are usually designed for specific task. They are not general enough and often omit parts of the features of the graph data. Meanwhile, many nodes in the graph also share similar information and interact via global attributes. But it has not been thoroughly considered by most previous works.

To overcome these limitations, we provide a definition of GNN and address the importance of global attribute in it. A GNN's nodes and edges have states, which could be updated through update functions, while the information between them could be conveyed via aggregate functions. Meanwhile, a global attribute is defined, which collects and updates information from all nodes/edges via global aggregate and update functions, while updating the states via global update function.

Under this, a concrete realization of GNN model called Graph-LSTM is proposed and used in SGG. The tag feature, a distribution depicting the probability of candidate semantic concepts, is generated from [8] and sent to the LSTMs as the global attribute via Semamtic Compositioanl Network (SCN). The Region Proposal Network (RPN) produces item features as initial states for node-LSTM and edge-LSTM, and after one update two LSTMs perform aggregate procedure via message passing. They iteratively update and aggregate until maximum iteration or convergence. And finally the hidden states of the LSTMs are input into inference modules to generate the triplets.

We evaluate the proposed Graph-LSTM model for SGG on Visual Genome (VG) [9] dataset. The R@50 recall score of triplet detection is reported and shows that the Graph-LSTM outperforms the Message Passing [10] and attention GCN [11], proving the effectiveness of the Graph-LSTM, especially the global attribute.

2. Related Work

2.1. Graph Neural Networks

Traditional machine learning methods [1] are designed to process Euclidean data structure and don't work well on graph data. Therefore, research on Graph neural network (GNN) has become increasingly popular.

Before the GNN concept has been brought up, many similar concepts and models have been discussed. The Graph Convolutional Neural Network (GCN) [5] introduces CNN that operates directly on graphs. Message-passing neural network (MPNN) [6] uses the message passing mechanism to pass information between graph nodes [10] [12]. And [7] proposed a non-local neural network (NLNN) which unifies several self-attention methods. Meanwhile, some works focus on providing definition of the GNN. [13] presents a building block called “graph network (GN)”.

2.2. Scene Graph Generation

Scene graph generation (SGG) is an important task in visual scene understanding. The goal is to find the triplet relations $(subject, predicate, object)$, e.g. $(window, on, train)$ in a given image, while all triplets forming a graph. Most previous works adopt the object-centric strategy, trying to infer relationships from all object pairs. They usually focus on improving visual features and reduce complexity by merging pair candidates [10] [12]. Some GNNs related methods have also been introduced, for instance, an attention graph convolutional network (aGCN) [11] is developed to update node and relationship representations by propagating context between nodes in candidate scene graphs.

3. Graph-LSTM: A GNN Model with Global Attribute

3.1. Definition of Graph Neural Network (GNN)

The term "graph" refers to the definition of a directed, attributed graph with a global attribute. A GNN is a specific form of graph with graph-structured data as input. Based on the GN block in [3], we explain the definition of GNN in Table 1. The edge e_k and the node v_i are represented by neuron cells/units. The global attribute u is shared by all nodes and edges. The information flow in the GNN is realized by three aggregate functions. For instance, the edge can aggregate information from the nodes it linked and the global attribute via the function ρ. And

Table 1: Definition of GNN and its realization as Graph-LSTM for scene graph generation.

	Graph neural network	
	general definition	***Graph-LSTM***
node	v_i	node-LSTM unit
edge	e_i	edge-LSTM unit
global attribute	u	tag feature
edge update	$e_k^{t+1} = \phi(e_k^t, u^t)$	LSTM update
node update	$v_i^{t+1} = \phi(v_i^t, u^t)$	LSTM update
global update	$u^{t+1} = \phi(u^t)$	—
edge aggregate	$e_k^t = \rho(v_{kn}^t, u^t)$	message passing
node aggregate	$v_i^t = \rho(e_{im}^t, u^t)$	message passing
global aggregate	$u^t = \rho(u^t, e_k^t, v_i^t)$	—

the individuals in the GNN can also update states through three update functions. For example, the k-th edge's state is updated via function ϕ with its current state e_k^t and global attribute u^t.

3.2. Graph-LSTM Model

A realization of GNN model called Graph-LSTM is proposed. The Graph-LSTM's basic component is the LSTM [14] units with semantic composition network (SCN) [8] mechanism to handle global attribute. LSTM units represent the nodes and edges respectively. The states of the LSTM units are the states of the nodes and edges, as they are updated via LSTM iterations. Message passing is used to pass information between node and edge as the aggregate function. More detailed explanations of Graph-LSTM and its application for scene graph generation will be given in the next section.

4. Graph-LSTM Model for Scene Graph Generation

As shown in 1, the subject and object are nodes (green) in the graph, while the predicate is a directed edge (orange) linking the $(subject, object)$ pair. We select LSTM unit to represent the nodes and edges, i.e., a node-LSTM unit for subject/obeject, while an edge-LSTM for predicate.

4.1. Initial States of Graph-LSTM

Firstly, the RPN will detect all meaningful items in the image, such as the helmet, producing the item features (visual feature vector with bounding box). The item features serve as the initial states f_i^v of the node-LSTM, representing the subject/object. Meanwhile, the union-box of the item features' boundingboxes is calculated and the union-box features are generated as the initial state $f_{i\to j}^e$ of the edge-LSTM, representing the predicate. Meanwhile, the tag feature is generated by the Tag Feature Generator (TFG) from [8], serving as the global attribute via SCN.

4.2. Update and Aggregate

The update of is performed by the update of the two LSTMs for one timestep. Then the aggregate is performed via message passing. A node message function computes message m_i of node i based on the hidden states from its outbound edge-LSTM $h_{i\to j}$, inbound edge-LSTM $h_{j\to i}$ and its current state h_i. And m_i is added to the node-LSTM's input at the current time t.

$$m_i = \sum_{j:i\to j} \sigma(\mathbf{v}_1^T[h_i, h_{i\to j}])h_{i\to j} + \sum_{j:j\to i} \sigma(\mathbf{v}_2^T[h_i, h_{j\to i}])h_{j\to i} \tag{1}$$

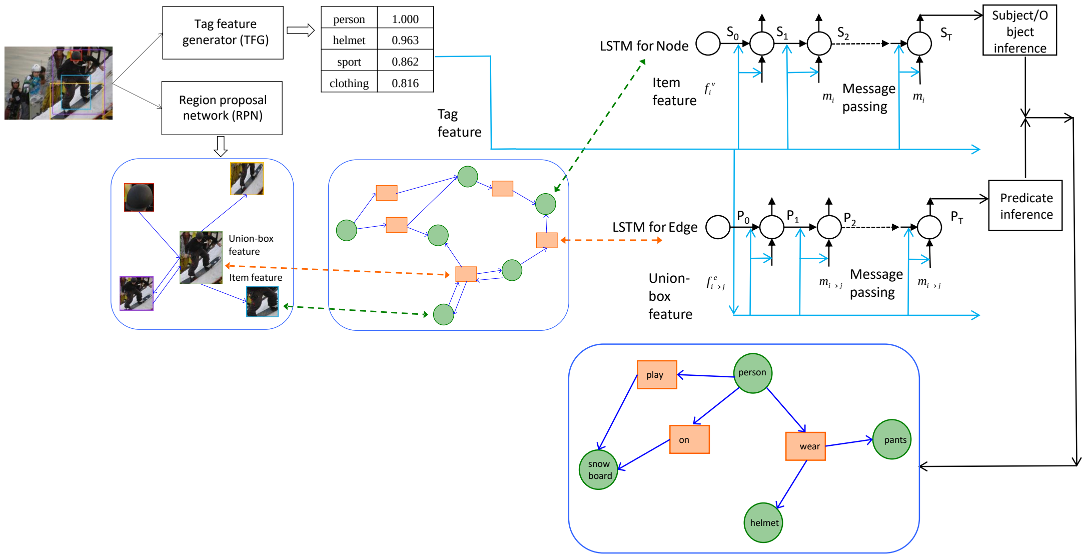

Figure 1: The RPN extracts visual features of nodes (green) and edges (orange) as the initial states of two LSTMs. A node message function and edge message function compute messages sent to the other LSTM as aggregate. Tag feature is generated by TFG and input into each timestep via SCN. Iteratively aggregate and update. Finally the inference modules output the triplet.

Similarly, an edge message function computes message $m_{i\to j}$ and feed it to the edge-LSTM.

$$m_{i\to j} = \sigma(\mathbf{w}_1^T[h_i, h_{i\to j}])h_i + \sigma(\mathbf{w}_2^T[h_j, h_{i\to j}])h_j \tag{2}$$

Perform the update and aggregate repeatedly until convergence or certain iterations. During these procedures, all training samples share the same node-LSTM and edge-LSTM weights respectively.

4.3. Semantic Compositional Network for Global Attribute

The global attribute provides valuable context information and interacts with each individual effectively. To the best of our knowledge, this is the first work that addresses the importance of global attribute and uses the tag feature as it for scene graph generation.

The tag feature is a distribution of all potential semantic concepts appearing in the image, noted as $\mathbf{s}$. $\mathbf{s}$ has a dimension of K (we adopt $K = 1000$ [8]), while K is the size of the tag vocabulary. s_k is the k-th element in $\mathbf{s}$, denoting the probability that the image contains this semantic concept. For instance, $(animal, 0.936)$ means the image has a probability of 0.936 for an animal. Each tag feature remains unchanged during computing.

We replace the traditional LSTM with the SCN-LSTM, which is basically training weighted sum (ensemble) of K LSTMs. The neural weights in LSTM can be rewritten as

$$\mathbf{W}_s = \sum_{k=1}^{K} s_k \mathbf{W}_\tau[k], \mathbf{U}_s = \sum_{k=1}^{K} s_k \mathbf{U}_\tau[k] \tag{3}$$

$\mathbf{W}_\mathcal{T}[k]$ and $\mathbf{U}_\mathcal{T}[k]$ denote the k-th 2D "slice" of $\mathbf{W}_\mathcal{T}$ and $\mathbf{U}_\mathcal{T}$. To reduce complexity, factorizing $\mathbf{W}(s)$ and $\mathbf{U}(s)$ is utilized [8].

$$\mathbf{W_s} = \mathbf{W}_a \cdot diag(\mathbf{W}_b\mathbf{s}) \cdot \mathbf{W}_c \tag{4}$$

$$\mathbf{U_s} = \mathbf{U}_a \cdot diag(\mathbf{U}_b\mathbf{s}) \cdot \mathbf{U}_c \tag{5}$$

Thus, the SCN-LSTM's update equations could be rewritten as following

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ia}\widetilde{x}_{i,t-1} + \mathbf{U}_{ia}\widetilde{h}_{i,t-1} + z) \tag{6}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{fa}\widetilde{x}_{f,t-1} + \mathbf{U}_{fa}\widetilde{h}_{f,t-1} + z) \tag{7}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{oa}\widetilde{x}_{o,t-1} + \mathbf{U}_{oa}\widetilde{h}_{o,t-1} + z) \tag{8}$$

$$\widetilde{\mathbf{c}}_t = \sigma(\mathbf{W}_{ca}\widetilde{x}_{c,t-1} + \mathbf{U}_{ca}\widetilde{h}_{c,t-1} + z) \tag{9}$$

$$\mathbf{c}_t = \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t + \mathbf{f}_t \odot \mathbf{c}_{t-1} \tag{10}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{11}$$

And for $\epsilon = i, f, o, c$, the equations are

$$\widetilde{\mathbf{x}}_{\epsilon,t-1} = \mathbf{W}_{\epsilon b}\widetilde{s} \odot \mathbf{W}_{\epsilon c}\widetilde{x}_{t-1} \tag{12}$$

$$\widetilde{\mathbf{h}}_{\epsilon,t-1} = \mathbf{U}_{\epsilon b}\widetilde{s} \odot \mathbf{U}_{\epsilon c}\widetilde{h}_{t-1} \tag{13}$$

4.4. Triplet Inference

After the last iteration, the hidden states of the two LSTMs are input into the subject/object inference and predicate inference modules, which are made of fully-connected layers. The triplet $(subject, predicate, object)$ will be generated by combining the inference results. And the scene graph is generated after inferring all potential feature candidates.

5. Experimental Results

We implemented and tested the Graph-LSTM in scene graph generation on Visual Genome (VG) [9] dataset. The VG dataset contains about 50000 training samples and 10000 testing samples, and the number of object categories is 150 while the number of predicate categories is 50. The R@50 recall score is reported, which measures the number of groundtruth relationship triplets that appear among the top 50 most confident triplet predictions in an image. A two-step training strategy is adopted. First fix the RPN and train the Graph-LSTM until convergence. Then fine-tune those two jointly. To make a more fair comparison, the implementation will adopt the same version of LSTM (or other recurrent neural networks) as the model to be compared with.

Table 2: Recall Score comparison with Message Passing [10] on Visual Genome dateset.

Scheme	Iteration Step	Recall@50
Message Passing[10]	—	8.55
Graph-LSTM	1	8.52
Graph-LSTM	2	8.76
Graph-LSTM	4	8.68

As shown in Table 2, the proposed Graph-LSTM outperforms the Message Passing scheme [10], achieving a 0.21 higher recall 50 score. And it demonstrates that 2 iterations provide the best performance. Since both schemes contain similar message passing, it's reasonable to conclude that the gain mainly comes from the global attribute.

As shown in Table 3, the proposed scheme outperforms the aGCN [11], and it significantly increases the score by 1.47. Similarly, 2 iterations provide the best performance.

Table 3: Recall Score comparison with aGCN [11] on Visual Genome dateset.

Scheme	Iteration Step	Recall@50
aGCN [11]	—	11.40
Graph-LSTM	1	11.79
Graph-LSTM	2	12.87
Graph-LSTM	4	12.38

6. Conclusions

Experimental results show that the proposed Graph-LSTM outperforms the Message Passing [10] and attention GCN [11], which proves the effectiveness of the proposed scheme. Considering the experiment settings and the scheme in [10] contains message passing similar to ours, the performance gain should mainly come from the global attribute, i.e., the tag feature. The tag feature is supposed to provide context information of the image, and guide the refinement of each item feature. And both experiments show best performance with 2 iterations.

Acknowledgments

This work was supported in part by National Science Foundation (CNS-1747783 and CCF-2007210) and Industrial Members of NSF Center for Big Learning (CBL).

References

[1] LeCun Y, Bengio Y and Hinton G 2015 *nature* **521** 436

[2] LeCun Y, Bengio Y *et al.* 1995 *The handbook of brain theory and neural networks* **3361** 1995

[3] He K, Zhang X, Ren S and Sun J 2016 *Proceedings of the IEEE conference on computer vision and pattern recognition* pp 770–778

[4] Erhan D, Szegedy C, Toshev A and Anguelov D 2014 *Proceedings of the IEEE conference on computer vision and pattern recognition* pp 2147–2154

[5] Duvenaud D K, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A and Adams R P 2015 *Advances in neural information processing systems* pp 2224–2232

[6] Gilmer J, Schoenholz S S, Riley P F, Vinyals O and Dahl G E 2017 *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (JMLR. org) pp 1263–1272

[7] Wang X, Girshick R, Gupta A and He K 2018 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 7794–7803

[8] Gan Z, Gan C, He X, Pu Y, Tran K, Gao J, Carin L and Deng L 2017 *Proceedings of the IEEE conference on computer vision and pattern recognition* pp 5630–5639

[9] Krishna R, Zhu Y, Groth O, Johnson J, Hata K, Kravitz J, Chen S, Kalantidis Y, Li L J, Shamma D A *et al.* 2017 *International Journal of Computer Vision* **123** 32–73

[10] Xu D, Zhu Y, Choy C B and Fei-Fei L 2017 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp 5410–5419

[11] Yang J, Lu J, Lee S, Batra D and Parikh D 2018 *Proceedings of the European conference on computer vision (ECCV)* pp 670–685

[12] Li Y, Ouyang W, Zhou B, Shi J, Zhang C and Wang X 2018 *Proceedings of the European Conference on Computer Vision (ECCV)* pp 335–351

[13] Hamilton W, Ying Z and Leskovec J 2017 *Advances in Neural Information Processing Systems* pp 1024–1034

[14] Sundermeyer M, Schlüter R and Ney H 2012 *Thirteenth annual conference of the international speech communication association*