J Inf Process Syst, Vol.16, No.6, pp.1250~1260, December 2020
https://doi.org/10.3745/JIPS.02.0147

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Multimodal Context Embedding for Scene Graph Generation

Gayoung Jung* and Incheol Kim**

Abstract

This study proposes a novel deep neural network model that can accurately detect objects and their relationships in an image and represent them as a scene graph. The proposed model utilizes several multimodal features, including linguistic features and visual context features, to accurately detect objects and relationships. In addition, in the proposed model, context features are embedded using graph neural networks to depict the dependencies between two related objects in the context feature vector. This study demonstrates the effectiveness of the proposed model through comparative experiments using the Visual Genome benchmark dataset.

Keywords

Deep Neural Network, Multimodal Context, Relationship Detection, Scene Graph Generation

1. Introduction

Scene graph generation task is a principal topic in artificial intelligence and computer vision that requires deep image understanding. Specifically, a scene graph represents a scene contained in an image, wherein nodes represent objects in the scene and edges correspond to the pairwise relationships between the objects. Accordingly, a scene graph can be considered as a fact set that expresses the visual relationships of the corresponding image as a triplet in the form <subject-relationship predicate-object>. That is, generating a scene graph involves generating a knowledge graph that represents the image scene based on the results from a deep understanding of an input image.

Fig. 1 shows a scene graph generation process. Generating a scene graph requires detecting objects and the relationship between objects in a specified image. Although object detection has been extensively studied in the field of computer vision, relationship detection has only recently received significant attention and is in its early stages of development. Several relationships exist between two objects in an image. In general, the two types of relationships that are frequently considered in scene graph generation studies are the spatial and semantic relationships. Whereas spatial relationship expresses the relative spatial relationship, such as “on,” “next to,” and “in front of,” between objects in an image, the semantic relationship represents associations, such as “wearing,” “eating,” and “holding,” related to the behavior of one object to another. Although object detection technologies using a convolutional neural network

Manuscript received June 26. 2020; first revision September 4, 2020; accepted September 14, 2020.

Corresponding Author: Incheol Kim (kic@kyonggi.ac.kr)
* Dept. of Computer Science, Graduate School of Kyonggi University, Suwon, Korea (jgyy4775@kyonggi.ac.kr)
** Dept. of Computer Science, Kyonggi University, Suwon, Korea (kic@kyonggi.ac.kr)

(CNN) achieve high performance, errors can occur in object classification and area detection. This indicates that there can be inaccuracies and errors in classifying two objects, which is the basic goal of object detection. Furthermore, even after acquiring a clear classification of two objects with relationships, accurately predicting the relationship between the objects can be difficult because of the large number of possible relationship types between the two objects. In general, several semantic constraints exist in relationships wherein two objects are related. For example, in Fig. 1, it is common to accept that the <man-wearing-shoes> relationship is rational, but the <man-wearing-racket> and <shoes-wearing-man> relationships are technically impossible. Thus, a consistent graph generation model should be developed to indicate the relationship characteristics to generate accurate scene graphs from images.

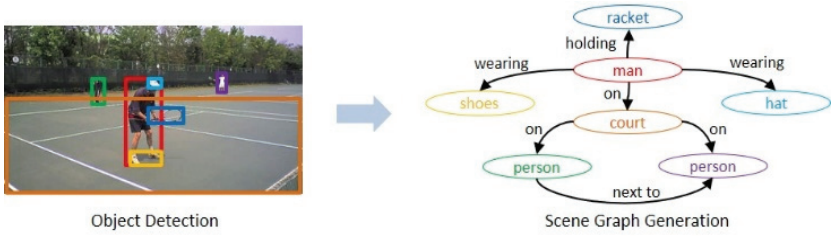


Fig. 1. Example of scene graph generation.

Several studies on scene graph generation [1,2] utilized visual features extracted from images through CNNs for object and relationship detection to generate scene graphs. In contrast, Yang et al. [2] developed a model using the attentional graph convolutional network (aGCN), a graph neural network technique that applies the context features of neighboring nodes of the scene graph to the feature values. Furthermore, in a study on visual relationships detection conducted by Liao et al. [3], natural language features, such as image caption and object category extracted from text, were used alternately to the visual features of an image extracted through CNN to accurately detect relationships between two objects. However, unlike scene graph generation, where object and relationship detection are performed simultaneously by indicating the interaction between two objects, the corresponding study focused on detecting only the relationship between objects based on pre-detected objects.

Based on these limitations, we propose a novel deep neural network model for scene graph generation. (1) To accurately detect objects and relationships, the proposed model utilizes several multimodal context features, including linguistic features and visual context features. (2) To indicate the order and role of each object in the expression of <subject-relationship predicate-object>, the model also uses a bidirectional recurrent neural network (biRNN) to generate linguistic context feature vectors. (3) Furthermore, the proposed model performs context embedding using a graph neural network to ensure that the dependencies between two objects in a relationship are fully considered in the graph node feature values. (4) We performed comparative experiments using the Visual Genome benchmark dataset [4] to demonstrate the accuracy and effectiveness of the proposed model.

2. Related Work

Visual relationship detection (VRD) is an image understanding task that detects two objects and classifies the relationship between them. Studies on visual relationship detection use object detectors,

such as Faster R-CNN. Thereafter, various features of each object region obtained as object detection results are used to detect the relationships between objects. Liao et al. [3] determined the relationships between two objects using the semantic similarity of words expressing the subjects and objects. In addition, Dai et al. [5] predicted relationships using visual features and positions of each object region (spatial feature) in an input image. Gkanatsios et al. [6] used visual features driven by a multimodal attentional mechanism that exploits spatio-linguistic similarities in a low-dimensional space. However, these studies on visual relationship detection focused only on detecting object relationships using pre-detected objects in an image. Thus, the performance of object detection can affect relationship detection; consequently, relationship detection cannot be independently improved. Recently, in addition to the VRD methods, several studies on scene graph generation have considered the interaction between object detection and relationship detection. The studies on scene graph generation have used message passing techniques to enable sending and receiving each context feature obtained by performing object detection and relationship detection simultaneously [1,2]. This method shares the advantage of ensuring the accuracy of object and relationship detection. Additionally, several message-passing methods have been developed to enable the exchange of context features. Popular methods include using an external knowledge base [7], using self-attention [8,9], and using graph convolutional networks (GCNs) [2].

A graph, which is a data structure composed of nodes and edges, is an effective tool for expressing relationships and interactions between objects. Social networks, knowledge graphs, and protein-protein interaction networks are typical examples of widely used graph data. Recently, owing to the high expressive power of such graphs, several studies on graph neural networks have been conducted to directly process these graphs. The graph neural networks can indicate dependencies between nodes through the message-passing edges between nodes in a graph. A GCN is a graph neural network designed to have each node of the graph aggregate feature values of neighboring nodes in a convolutional fashion [10]. Recently, GCNs have been used in artificial intelligence fields, such as scene graph generation, visual question answering, and visual commonsense reasoning. In this study, we used eGCN [2] to extract accurate context features to detect relationships between objects for scene graph generation. In the study by Yang et al. [2], the authors proposed an eGCN that extends the traditional GCN, wherein message-passing was conducted in one direction along directional edges, to enable message-passing in two directions.

3. Scene Graph Generation Model

3.1 Model Overview

Fig. 2 shows the neural network structure of the proposed scene graph generation model, which comprises three main stages: region proposal (RP), object and relationship detection (ORD), and graph generation (GG). In the RP stage, Faster R-CNN, a widely used object detection module, is used. Furthermore, for each object candidate region of the input image, the ResNet 101 visual feature vector, bounding box position and size, and object class distribution are determined. The ORD stage consists of three main sub-stages: graph initialization, graph reasoning, and graph labeling. In graph initialization, object and relationship nodes for generating a scene graph are constructed using the object region results from the preceding RP stage. Thereafter, initial feature values are assigned to these nodes. In graph

reasoning, context features are exchanged between neighboring object and relationship nodes in the graph through the GCN, and the feature values of each node are updated. In graph labeling, based on the final feature values of each node, node classification is performed for objects and relationships. Finally, in GG, which is the final step of the proposed model, a single scene graph is generated based on each classified node.

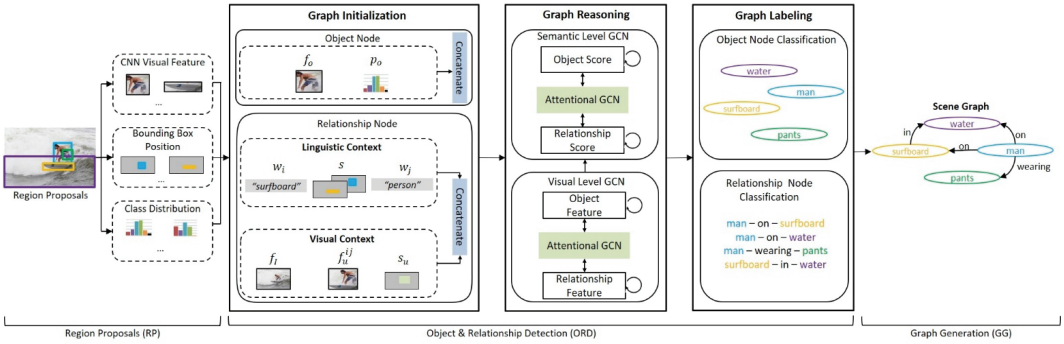


Fig. 2. Structure of the proposed method.

3.2 Object Node Features

In the graph initialization stage of the proposed method, one object node is generated in the graph for each object region detected in the image, and an initial feature value is assigned to each corresponding node. The proposed model applies Faster R-CNN to the input image to assign the initial feature values of each object using the visual feature vector and object class probability distribution extracted from each object candidate region. The initial feature values are used for object node classification after being combined with the context features of neighboring nodes through a graph neural network. Accordingly, the final object category of each node classified by the proposed model can differ from the initial object category estimated by Faster R-CNN.

- **Object visual feature**
  - f_o: CNN-based visual features of the object region
- **Class probability distribution**
  - p_o: Object class probability distribution of object region

Eq. (1) expresses the initial feature vector O of each object node.

$$O = [f_o, p_o] \tag{1}$$

3.3 Relationship Node Features

In the graph initialization stage, initializations of the object and relationship nodes are performed as described earlier. That is, a relationship node is generated for each pair of object regions detected in the input image, and each relationship node is assigned an initial feature value. Unlike existing models, to achieve accurate relationship detection, the proposed model provides detailed multimodal context features that include text-based linguistic and image-based visual context features as initial feature values

of the relationship nodes. Details of the visual and linguistic context feature sets for relationship nodes are as follows:

- **Visual context feature set**
  - f_I: Convolutional visual features of the input image
  - f_u^{ij}: Convolutional visual features of a union box around the subject region and object region that can form a relationship
  - s_u: Position feature of the union box around the subject and object.

$$s_u = (\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{(x_{br}-x_{tl})(y_{br}-y_{tl})}{WH}) \tag{2}$$

In Eq. (2), x, y, w, h denote the center coordinates, width, and height of the object region, respectively. Additionally, W, H represent the width and height of the union box, respectively. Moreover, in Eq. (3), x_{tl}, y_{tl} and x_{br}, y_{br} represent the upper left corner and lower right corner coordinates of the union box, respectively.

- **Linguistic context feature set**
  - w_i: Embedding of the predicted object category for the specified subject using multi-layer perceptron (MLP)
  - s: Position features of the subject and object regions in the specified image
  - w_j: Embedding the predicted category of the object using MLP

$$s = [x_i, y_i, w_i, h_i, \frac{x_i - x_j}{W}, \frac{y_i - y_j}{H}, log\frac{w_i}{w_j}, log\frac{h_i}{h_j}, x_j, y_j, w_j, h_j] \tag{3}$$

By combining these three elements, w_i, s, and w_j, the linguistic feature vector y for expressing a relationship can be determined by combining methods, such as a unidirectional RNN, biRNN, and concatenation. In general, the optimal method to express a relationship between two objects is through a single sequence, such as <subject-relationship predicate-object> after indicating the position, order, and role of the three linguistic features. Accordingly, the proposed model generates linguistic context vectors y by sequentially combining the three linguistic features (w_i, s, w_j) using a biRNN. In particular, based on the linguistic conceptual relationship, to effectively indicate the bidirectional constraints between the subject and object in the feature vector y, the proposed model used biRNN for the embedding of the linguistic context sequence, $< w_i, s, w_j >$. Fig. 3 shows the biRNN-based linguistic context feature embedding process, and Eq. (4) expresses this process mathematically.

$$y = (W_{\vec{h}_{y_1}}\overrightarrow{h_1} + W_{\overleftarrow{h}_{y_1}}\overleftarrow{h_1}) + (W_{\vec{h}_{y_2}}\overrightarrow{h_2} + W_{\overleftarrow{h}_{y_2}}\overleftarrow{h_2}) + (W_{\vec{h}_{y_3}}\overrightarrow{h_3} + W_{\overleftarrow{h}_{y_3}}\overleftarrow{h_3}) \tag{4}$$

W denotes a training parameter, $\vec{h}$ represents a hidden state in the forward direction, and $\overleftarrow{h}$ represents a hidden state in the backward direction. In the proposed model, the initial feature values of each relationship node are determined by combining the visual and linguistic context feature vectors embedded with the biRNN, as shown in Eq. (5).

$$R = [f_I, f_u^{ij}, s_u, y] \tag{5}$$

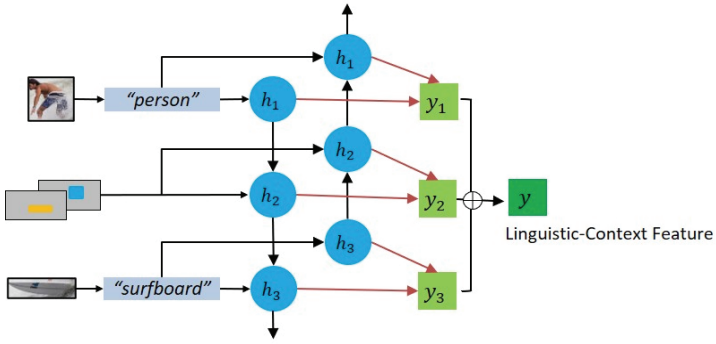


Fig. 3. biRNN-based linguistic context feature embedding.

3.4 Graph Reasoning and Labeling

The graph reasoning process of the proposed model comprises two levels of GCNs: visual reasoning and semantic reasoning levels. At each level, based on the initial feature values assigned to each node in the graph initialization stage, context features are exchanged between neighboring nodes in the graph. Thereafter, the feature values of each node are updated. In particular, the proposed model uses aGCN to classify nodes that require and that do not require attention among the neighboring nodes, and applies neighboring node features in the feature value update process of each node. The attention values a_i of each node are predicted based on the feature values of z_i and z_j of the two nodes, as shown in Eqs. (6) and (7).

$$m_{ij} = \omega\sigma(W_a[z_i^{(l)}, z_j^{(l)}]) \tag{6}$$

$$a_i = softmax(m_i) \tag{7}$$

In Eqs. (6) and (7), σ denotes a two-layer perceptron (MLP), and ω and W represent training parameters.

When updating the feature values of object nodes using the aGCN, context features are exchanged in the following node relationships: subject node<->object node, subject node<->relationship node, and object node<->relationship node. However, when updating the feature values of relationship nodes, context features are exchanged in two relationships: relationship node<->subject node and relationship node<->object node. Accordingly, Eq. (8) expresses the feature value update of each object node in the graph, and Eq. (9) expresses the feature value update of each relationship node.

$$z_i^o = \sigma(W_{so}Z^o a_{so} + W_{sr}Z^r a_{sr} + W_{or}Z^r a_{or}) \tag{8}$$

$$z_i^r = \sigma(z_i^r + W_{rs}Z^o a_{rs} + W_{ro}Z^o a_{ro}) \tag{9}$$

In Eqs. (8) and (9), s, r, and o represent subject, relationship, and object nodes, respectively. Using these equations, node feature value update processes are performed at the two aGCN levels, visual reasoning, and semantic reasoning levels. Furthermore, the object and relationship class probability distributions of each node from the visual reasoning level are used as the initial node input in the semantic reasoning level.

Finally, in the graph labeling stage, objects and relationships are classified based on the final feature values of each node acquired from the semantic graph reasoning stage. The object nodes are labeled based

on the largest value in the object class probability distribution. The relationship nodes are also labeled through the same process. Using the labeling process, standardized results are obtained in tuples of the form <object-predicate-subject>.

4. Implementation and Experiment

In this study, we used the Visual Genome [4] benchmark dataset to examine the performance of the proposed model. This dataset is one of the most commonly used datasets in scene graph generation. It consists of a total of 108,077 images, and each image is labeled with the objects and their relationships. Without using all the object and relationship types in the image, 150 top-frequency object classes and 50 relationship classes were selected and used for training and experiment. As a result of the preprocessing, approximately 82,000 images remained, and each image had approximately 11.5 objects and 6.2 relationships in the scene graph. Among these images, 56,224 were used for training and 26,446 were used as tests.

The proposed model was implemented in the Ubuntu 16.04 LTS environment using PyTorch, a Python deep learning library. The model training and test processes were performed using a hardware with a GeForce GTX 1080Ti GPU. For the model training process, the batch size was set to 8, whereas the epoch was set to 6. Additionally, the learning rate was set to 0.005, and stochastic gradient descent was used as the optimizer.

To examine the performance of the proposed scene graph generation model, three scene graph generation evaluation metrics were used: SGGen (Scene Graph Generation), PhrCls (Phrase Classification), and PredCls (Predicate Classification). These metrics differ in the degree of ground truth used. The SGGen metric involves predicting all locations, labels, and relationships of the object, whereas PhrCls involves using the ground truth object locations to predict labels and relationships. The PredCls method uses the ground truth object locations and labels to predict only the relationships. In all metrics, the recall of triplet sequences representing the scene graph was measured. The choice of this metric is because of the sparsity of the relationship annotations in Visual Genome. All components of the triplet, namely, subject, object, and relational predicate, should be equal to the ground truth to be identified as ground truth. Additionally, the SGGen metric identifies the triplet as the ground truth if two object locations are equal to the ground truth locations and have an intersection over union (IoU) value of at least 0.5. The experiment was conducted with the three metrics by computing the recall at the top 50 (r@50) and 100 (r@100) to measure the performance.

In the first experiment, we evaluated the effectiveness of biRNN, which was used as the linguistic context feature embedding method for the relationship nodes in the proposed method. This experiment compared the biRNN embedding method with two other embedding methods: RNN and simple concatenation of linguistic context features.

Table 1 shows the experimental results of the comparison of the embedding methods. From Table 1, the biRNN method achieved the highest performance in all three evaluation metrics. Additionally, the concatenation method yielded the lowest performance because it did not apply the sequence of <subject-predicate-object>. Although the RNN-based embedding method showed better results compared to the concatenation method because it applied the sequence features, it obtained lower performance results compared to the biRNN-based embedding method, which uses the bidirectional feature.

Table 1. Performance comparison of linguistic context feature embedding methods

Method	SGGen		PhrCls		PredCls	
	r@50	r@100	r@50	r@100	r@50	r@100
Concat	24.35	27.07	41.96	52.50	65.20	69.32
RNN	24.82	27.20	43.33	53.81	65.98	69.75
biRNN	24.91	27.61	43.69	54.16	66.87	71.15

In the second experiment, we analyzed the effects of multimodal relationship node feature values, including the linguistic context feature set (LC) and visual context feature set (VC). This experiment compared the three cases as follows: using only VC, using only LC, and using both feature sets. In these cases, the same object node feature values were used. Additionally, for LC, the biRNN-based embedding method was used.

Table 2 displays the experimental results of comparing feature sets. Notably, the highest performance was obtained when both LC and VC were used for the relationship node feature values, as in the proposed model. In addition, compared to using only the VC, the case of using only LC exhibited a relatively higher performance. Based on these results, we can assume that VC is more effective in improving performance than LC.

Table 2. Performance comparison of feature sets

Feature Set		SGGen		PhrCls		PredCls	
VC	LC	r@50	r@100	r@50	r@100	r@50	r@100
∨		23.93	26.92	40.50	51.30	63.94	68.35
	∨	24.49	27.35	42.76	53.29	66.92	71.03
∨	∨	24.91	27.61	43.69	54.16	66.87	71.15

In the third experiment, we compared the performance of our proposed scene graph generation model with other state-of-the-art models. As described earlier, models 1 and 2 use only visual features, whereas model [3] uses linguistic features obtained from image captions. Furthermore, unlike models 1 and 3, model 2 includes the graph neural network-based node feature value embedding process. As shown in Table 3, the proposed model outperformed the state-of-the-art models in the SGGen and PhrCls metrics. However, in the PredCls metric, model 3 outperformed the proposed model. Unlike the proposed model, model 3 uses caption text data of the input image, which may have resulted in better performance. However, additional text descriptions other than the image are rarely provided in a typical scene graph generation task. Thus, from the results, we can conclude that the proposed model yields optimal performance for a scene graph generation task.

Table 3. Performance comparison of scene graph generation models

	SGGen		PhrCls		PredCls	
	r@50	r@100	r@50	r@100	r@50	r@100
Model 1	10.72	14.22	24.34	26.50	67.03	71.01
Model 2	11.40	13.70	29.60	31.60	54.20	59.10
Model 3	22.17	23.62	28.58	31.69	85.02	91.77
Proposed model	24.91	27.61	43.69	54.16	66.87	71.15

In the last experiment, we compared and analyzed the performances of the proposed model on relationship detection for each relational predicate. In this experiment, several spatial and semantic relationships with the top frequency in the Visual Genome dataset were selected. The selected spatial relationships were "on," "near," and "in," and the selected semantic relationships were "has," "riding," "holding," "sitting on", and "wearing."

Table 4 lists the experiment results of comparing the predicate detections. In the proposed model, the "on" and "near" relationships, which can be easily distinguished among the three spatial relationships, showed high detection rates of at least 96% on both r@50 and r@100 recall scales. However, the "in" relationship, which is relatively difficult to distinguish because only a part of the object can be exposed, showed a detection rate slightly below 90%. Furthermore, for the semantic relationships, the "has," "riding," and "wearing" relationships showed high detection rates of at least 90% on both r@50 and r@100 recall scales. However, the "holding" and "sitting on" relationships showed low detection rates that were below 80%. This result may be attributable to the "holding" and "sitting on" relationships appearing less frequently in the dataset compared to other relationships. Additionally, the "holding" and "sitting on" relationships share similar object location characteristics with "has" and "on" relationships, respectively; hence, the detection results may have been less accurate.

Table 4. Performance comparison of predicate detections

Predicate	r@50	r@100
on	97.22	97.73
near	96.95	97.41
in	88.50	89.23
has	96.78	97.20
riding	90.72	91.88
wearing	98.53	99.02
sitting on	49.17	50.87
holding	78.35	79.95

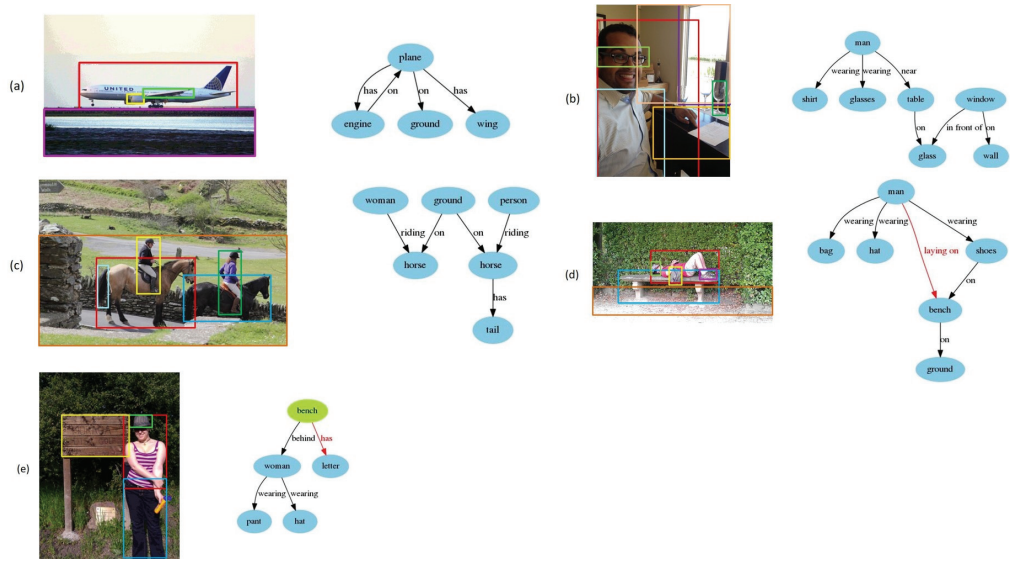


Fig. 4. Scene graph generation results of the proposed model: (a, b) no errors in either object or relationship detection, (c) some errors in object detection, (d) some errors in relationship detection, and (e) some errors in both object and relationship detection.

To conduct a qualitative evaluation of the performance of the proposed model, we examined several scene graph examples generated from the Visual Genome benchmark dataset using the proposed model. The images on the left of Fig. 4 display the input images and object regions detected by the proposed model for each image. Furthermore, the images on the right display the scene graphs generated by the proposed model for each input image. Fig. 4(a) and 4(b) are examples where the proposed model successfully generated appropriate scene graphs from the specified images. From the figure, object regions of various sizes in the images were accurately detected. Moreover, based on the object detection results, the relationships between the objects were accurately determined to generate the final precise scene graph. In Fig. 4(c), although the proposed model was able to accurately detect the main objects, such as “person,” “woman,” and “horse,” during the object detection stage, additional minor objects or parts, such as “helmet,” and “tail,” were undetected. Consequently, the proposed model also failed to detect relationships, such as “wearing” and “has,” which are associated with undetected objects. In Fig. 4(d), although the proposed model accurately detected objects included in the image, such as “man,” “bag,” “hat,” and “bench,” detecting the <man-laying on-bench> relationship among the object relationships failed in the final generated scene graph. This could be because of insufficient “laying on” relationship cases in the training dataset. Moreover, in Fig. 4(e), the proposed model incorrectly detected the object “sign” in the image as “bench.” Consequently, the generated scene graph shows an incorrect relationship of <bench-has-letter>, instead of the correct relationship, <sign-on-letter>. Based on the findings in Fig. 4, we propose additional studies on improving the accuracy of the object detection module, which is a vital component of scene graph generation, and solving the imbalance of training data to improve the performance of the proposed model.

5. Conclusion

This study proposes a model using a deep neural network for accurate scene graph generation of an image. The proposed model uses multimodal context features, such as visual and linguistic context features. In particular, a separate biRNN network was used to maximize the effects of linguistic context features. In addition, the model included context feature embedding using a graph neural network to accurately indicate the dependencies between two related objects in the graph node feature values. Furthermore, the performance superiority of the proposed model over the state-of-the-art models was verified by performing comparison experiments using the Visual Genome benchmark dataset. In the future, we will investigate further structures that can improve the performance of the proposed model by enhancing the accuracy of the object detection module and resolving the imbalance problem of the training data.

Acknowledgement

This research was supported by the Ministry of Science and ICT, Korea, under the Information Technology Research Center support program (No. IITP-2017-0-01642) supervised by the Institute for Information & Communications Technology Promotion (IITP).

References

[1] Y. Li, W. Ouyang, B. Zhou, K. Wang, and X. Wang, "Scene graph generation from objects, phrases and region captions," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 1261-1270.

[2] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph R-CNN for scene graph generation," in *Proceedings of the 15th European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 670-685.

[3] W. Liao, B. Rosenhahn, L. Shuai, M. Y. Yang, "Natural language guided visual relationship detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, 2019, pp. 444-453.

[4] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, et al., "Visual Genome: connecting language and vision using crowdsourced dense image annotations," *International Journal Of Computer Vision*, vol. 123, no. 1, pp. 32-73, 2017.

[5] B. Dai, Y. Zhang, and D. Lin, "Detecting visual relationships with deep relational networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, 2017, pp. 3076-3086.

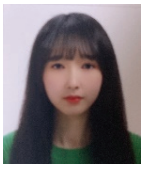
[6] N. Gkanatsios, V. Pitsikalis, P. Koutras, A. Zlatintsi, and P. Maragos, "Deeply supervised multimodal attentional translation embeddings for visual relationship detection," in *Proceedings of 2019 IEEE International Conference on Image Processing*, Taipei, Taiwan, 2019, pp. 1840-1844.

[7] J. Gu, H. Zhao, Z. Lin, S. Li, J. Cai, and M. Ling, "Scene graph generation with external knowledge and image reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 1969-1978.

[8] M. Qi, W. Li, Z. Yang, Y. Wang, and J. Luo, "Attentive relational networks for mapping images to scene graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, 2019, pp. 3957-3966.

[9] S. Woo, D. Kim, D. Cho, and I. S. Kweon, "LinkNet: relational embedding for scene graph," *Advances in Neural Information Processing Systems*, vol. 31, pp. 560-570, 2018.

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, Toulon, France, 2017.



Gayoung Jung **https://orcid.org/0000-0002-7314-8108**

She received a B.S. degree in Computer Science from Kyonggi University in 2019. She is currently an M.S. student of the Department of Computer Science, Kyonggi University, Korea. Her current research interests include machine learning, computer vision, and intelligent robotic systems.

Incheol Kim **https://orcid.org/0000-0002-5754-133X**

He received a B.S. degree in mathematics, an M.S. degree in computer science, and a Ph.D. degree in computer sciences in 1985, 1987, and 1995, respectively, from Seoul National University. He is currently a professor in the Department of Computer Science, Kyonggi University, Korea. His research interests include artificial intelligence and intelligent robotic systems.