GANG: Guided Attention Network in Graphs for VQA

Minseo Yoon
Data Science
****
*********@korea.ac.kr

Jiwon Jeong
Data Science
*2021320****
jjwon4086@korea.ac.kr

Jaewon Min
Computer Science and Engineering
****
*******@korea.ac.kr

***Abstract*—Combining scene graph generation with visual question answering has been actively studied and developed recently. There is a problem that the need for exploring many parts of scene graph reasoning limits more flexible reasoning. In this paper, we propose an approach to leverage a specific object in the question that corresponds image objects. We utilize this procedure by guiding attention and combining attention prior with attention score. And we propose a model that conducts this process, Guided Attention Network in Graphs (GANG). We show that our approach outperforms the baseline model (GraphVQA) and the performance of GANG is boosted by GAT-like encoding. Our code is available at https://github.com/stop1one/GANG-VQA**

1. Introduction

Computer vision technology has greatly improved the ability of machines to understand visual data. Among the various tasks in computer vision, scene understanding and visual question answering (VQA) have received considerable attention due to their real-world applications and potential to bridge the gap between human and machine perception. Scene graph generation and visual question answering are two fundamental tasks that enable a comprehensive understanding of images and allow machines to make inferences about visual content.

Scene graph generation aims to extract a structured representation from an image that captures the relationships between objects. Objects are represented as nodes and relationships as edges. This hierarchical representation provides a rich contextual understanding of the visual scene, allowing for advanced reasoning and interpretation of the image. Scene graphs have been introduced for image retrieval [6], image generation [5], image captioning [2], understanding instructional videos [4], and situational role classification [10].

Visual question answering focuses on developing algorithms that can answer natural language questions about images. Given an image and a natural language question about the image, the task is to provide an accurate natural language answer [1]. VQA systems combine visual recognition and language understanding to enable machines to understand and respond to human-like queries related to visual content.

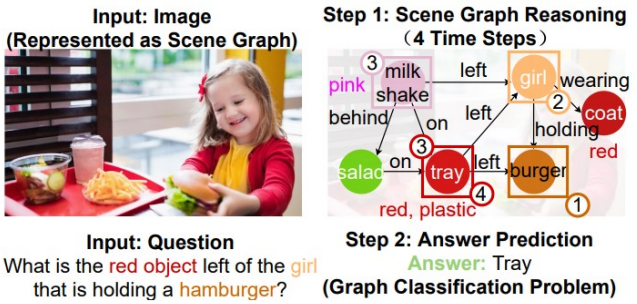

Figure 1. **Scene graph**: Scene graph encodes objects (e.g., girl, burger) as nodes connected via pairwise relationships (e.g., holding) as edges. **Baseline Framework**: Translating and executing a natural language question as multiple iterations of message passing among graph nodes (e.g., hamburger → small girl → red tray). The final state after message passing represents the answer (e.g., tray).

The ultimate goal of VQA is to build models that reason about their relationships, attributes, and behavior in response to various questions. By combining scene graph generation with VQA, we can leverage the complementary strengths of both approaches. The scene graph provides a structured representation of visual scenes, offering a solid foundation for reasoning and answering questions. Conversely, VQA brings natural language understanding and reasoning capabilities, allowing for more expressive and interactive image understanding.

To support question answering on scene graphs, we propose Guided Attention Network in Graphs (GANG). From the perspective of the association between the scene graph and the question, we present an insight into a guidable structure. By a graph neural network (GNN) layer, an image encoder and relation encoder can construct a GNN framework for Scene Graph Question Answering (Scene Graph QA). As shown in Figure 1, the problem of exploring many parts of scene graph reasoning can be expected to be improved efficient reasoning and performance by directly guiding.

In this paper, we propose an approach to leverage a specific object in the question that corresponds image objects. Our method leverages the scene graph's rich contextual information to improve the accuracy and robustness

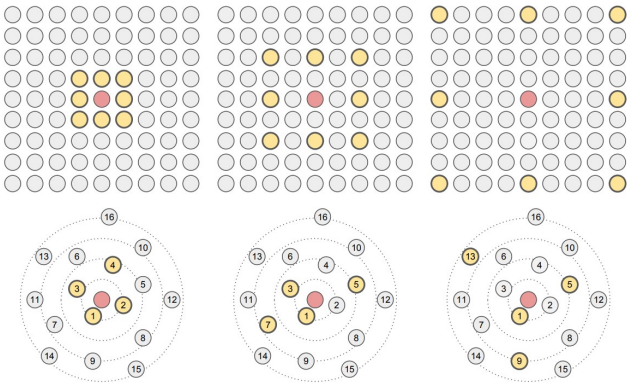

Figure 2. Visualization of dilated convolution on a structured graph arranged in a grid (e.g. 2D image) and on a general structured graph. (top) 2D convolution with kernel size 3 and dilation rate 1, 2, 4 (left to right). (bottom) Dynamic graph convolution with dilation rate 1, 2, 4 (left to right). This figure is taken from [9].

of VQA models. We evaluate our proposed approach on GQA datasets [3] and compare it with existing methods for scene graph generation and VQA. Our experimental results demonstrate that GANG can outperform the baseline model [11] by a large margin (49.60% vs. 46.21%). Our results suggest the importance of guided attention structure and attention prior information.

2. Related Work

These are the related architectures and ideas used in the experiment based on implementing GANG.

2.1. Graph Convolutional Networks (GCN)

GCNs [7] are a class of deep learning models specifically designed to operate on graph-structured data. GCNs extend the concept of convolutional neural networks (CNNs) to handle graph data, which can capture complex relationships and dependencies between entities. At the core of GCNs is the idea of neighborhood aggregation, which enables the model to gather information from a node's local neighborhood. Unlike regular convolutional layers in CNNs, which operate on regular grid-like data such as images, GCNs generalize the convolutional operation to work on irregular, non-Euclidean domains represented as graphs.

$$h_i^{(L)} = \sigma(\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (W_{GCN}^{(L)} \hat{h}_j^{(L-1)}))$$

$\hat{h}_j^{(L-1)}$ denotes the node feature as inputs to the L^{th} GNN layer.

The basic operation in a GCN involves updating the node representations by aggregating information from its neighboring nodes. This is achieved by taking a weighted sum of the feature vectors of neighboring nodes and the node itself. The weights are typically determined by the graph structure or learned through training. By iteratively applying this aggregation process, GCNs can capture and propagate information through the graph, allowing each node to have access to the collective knowledge of its local neighborhood.

2.2. Dilated Convolution in GCNs

Dilation-based GCNs [9] propagate information more slowly as distance increases. This allows embeddings to better understand the information on the entire graph by considering the structure of the graph.

The dilation-based approach introduces a dilation factor, which determines the spacing between nodes during message passing. In traditional GCNs, each node aggregates information from its immediate neighbors. However, in dilation-based GCNs, nodes skip some of their neighbors based on the dilation factor. This means that information from distant nodes is incorporated into the aggregation process but with a reduced frequency compared to neighboring nodes.

2.3. Graph Attention Networks (GAT)

GATs [13] are a type of graph neural network (GNN) that have been introduced to address the limitations of traditional graph convolutional networks (GCNs) in capturing important node relationships. GATs incorporate attention mechanisms into the graph convolution operation, allowing nodes to dynamically adjust the importance of their neighbors during information aggregation.

The key idea behind GATs is to assign attention weights to each neighbor node based on its relevance to the central node. This is achieved by computing an attention coefficient for each neighbor using a shared attention mechanism. The attention coefficients are computed by applying a learned compatibility function between the central node and its neighbors. By using attention mechanisms, GATs can automatically identify the most relevant neighbors and assign higher weights to them while downplaying the influence of less informative nodes.

$$\alpha_{ij}^{(L)} = \text{Softmax}_{\mathcal{N}_i}(MLP(\hat{h}_i^{(L-1)}, \hat{h}_j^{(L-1)}, \hat{e}_{ij}^{(L-1)}))$$
$$h_i^{(L)} = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(L)} W_{GAT}^{(L)} \hat{h}_j^{(L-1)})$$

Specifically, the attention score $\alpha_{ij}^{(L)}$ for message passing from node j to node i at L^{th} is calculated. $\text{Softmax}_{\mathcal{N}_i}$ is a normalization to ensure that the attention scores from one node to its neighbor nodes sum to 1. After calculating the attention scores, we calculate each node's new representation as a weighted average from its neighbor nodes. $\hat{e}_{ij}^{(L-1)}$ denotes the edge feature as inputs to the L^{th} layer.

2.4. Grounding-based Attention Priors (GAP)

GAPs [12] leverage the pairing of questions and images as a weakly supervised signal to learn the relationship

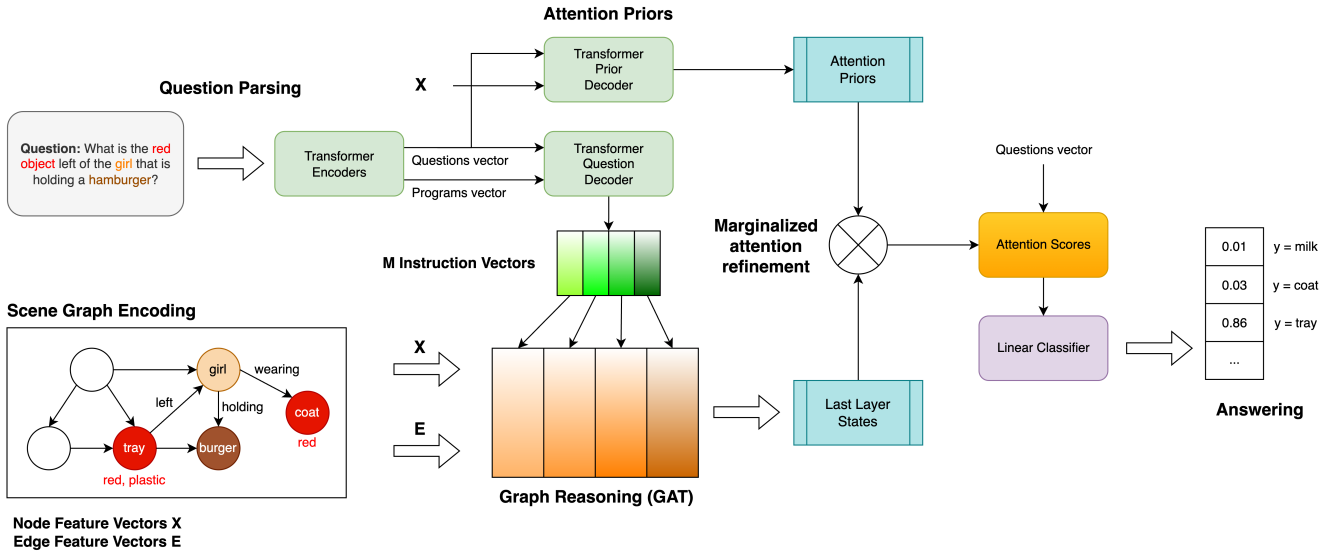

Figure 3. Semantics of the GANG Framework.

between words and image regions, eliminating the need for additional annotation. To address the challenge of disparate semantic interpretations between query words and image regions, GAPs construct a parse tree of the query, extract nested phrasal expressions, and ground them to image regions. These expressions align more semantically with image regions than individual words, providing more reliable linguistic-visual alignments.

The main goal of GAPs is to use these discovered alignments to guide reasoning attention. This guidance process consists of two complementary pathways. Firstly, attention weights are pretrained to align with the pre-computed grounding, achieved in an unsupervised manner without access to answer ground truths. Secondly, the attention prior is used to directly regulate and refine the attention weights based on the ground truth answer through back-propagation, ensuring they do not deviate significantly. A learnable gate modulates this process. These dual guidance pathways represent a significant advancement compared to previous attention regularization methods [12], [14] as they leverage linguistic-visual compatibility directly and flexibly during both training and inference, rather than solely for regularization purposes. This method is “marginalized attention refinement”.

3. Methods

We now present several approaches that we attempted to guide the answering process. Firstly, we introduce the Direct Guiding Answering method, where we directly compare the question with the scene graph to use the words in the question associated with scene graph. Secondly, we go beyond directly specifying words and introduce the Guided Attention Network in Graphs (GANG) model, which learns and incorporates the association between the scene graph and the question into the answering process.

3.1. Scene Graph Encoding

This is just the method that we will use for performance boosting before we get the final result. We implemented and experimented with GCN-based, Dilation GCN-based, and GAT-based models to measure in which method the performance improvement with scene graph reasoning occurs the most. Among them, the accuracy of the GAT-based model was recorded as the highest, so we decided on GAT-based encoding as a model for the final performance boosting.

The baseline is MLP-based encoding, and for the simplicity of experiments and models, our models are tested first based on MLP. After that, we used our own GAT-based encoding to determine to for how much performance can increase. Detailed results can be seen in section 4.

3.2. Directly Guiding Answering (DGA)

In the existing approach [11], the instruction vector obtained through the Seq2Seq module from the question is concatenated with the node features and edge features of the scene graph. The concatenated vector is then used as input to the GNN layer.

$$\mathbf{i} = [\mathbf{i}^{(1)}, ..., \mathbf{i}^{(M)}] = \text{Seq2Seq}(q_1, ..., q_Q)$$
$$\hat{\mathbf{h}_i}^{(L-1)} = [\mathbf{h}_i^{(L-1)}; \mathbf{i}^{(L)}]$$
$$\hat{\mathbf{e}_{ij}}^{(L-1)} = [\mathbf{e}_{ij}^{(L-1)}; \mathbf{i}^{(L)}]$$

where $(q_1, ..., q_Q)$ denotes question vector, $\mathbf{i}$ denotes instruction vector, $\mathbf{h}_i^{(L-1)}$ and $\mathbf{h}_{ij}^{(L-1)}$ denote the node feature and edge feature of inputs to the L^{th} GNN layer.

However, the instruction vector contains the entire sentence of the question, which includes many words that are unrelated to the scene graph (e.g., ”the,” ”a,” etc.).

TABLE 1. EVALUATION RESULTS ON GQA

Method	Binary	Open	Consistency	Validity	Plausibility	Distribution	Accuracy
Baseline (GraphVQA + MLP)	53.13	39.86	75.00	93.24	85.65	0.60	46.21
GraphVQA + GCN	54.15	37.29	75.38	93.49	85.75	0.69	45.36
GraphVQA + GCN (Dilation = 2)	54.54	36.38	75.00	**93.64**	86.16	0.70	45.07
GraphVQA + GAT	**56.71**	40.17	**80.33**	92.99	86.34	**0.50**	48.09
DGA + MLP	54.67	36.64	72.58	93.05	86.10	0.55	45.27
GANG + MLP	55.48	41.25	75.00	93.24	86.06	0.59	48.06
GANG + GAT ($\lambda = 0.1$)	56.58	**43.19**	74.67	93.42	86.72	0.51	**49.60**
GANG + GAT ($\lambda = 0.2$)	55.74	40.91	63.33	93.53	86.94	0.57	48.01
GANG + GAT ($\lambda = 0.5$)	54.36	32.45	59.62	93.26	**87.33**	0.98	42.94

Therefore, we attempted to directly concatenate and guide only the words that include the objects from the scene graph.

$$\mathbf{i}' = [\mathbf{i}'^{(1)}, ..., \mathbf{i}'^{(M)}] = \text{Seq2Seq}(q_1, \text{pad}, q_3, ..., \text{pad}, ..., q_Q)$$

Before creating the instruction vector, we first compare the scene graph with the question sentence. We extract only the words from the question within the scene graph, while replacing all other words with pad token. The modified question vector is then passed through the Seq2Seq module to generate the guided instruction vector $\mathbf{i}'$.

3.3. Guided Attention Network

Inspired by the DGA method mentioned above, we devised an approach that goes beyond directly selecting words from the question. Instead, we introduced a trainable network to facilitate the association between the scene graph and the question. To learn such associations and guide the answering process directly, we computed the Attention Prior and incorporated it into the results of the GNN.

The most crucial component in the GANG model is the Transformer responsible for computing the Attention Prior. The Transformer encoder used in the conventional question parsing process remains unchanged, while a new architecture called the *TransformerPriorDecoder* has been devised for the decoder component.

The Decoder-computed Attention Prior values and the output of the GNN are combined though "marginalized attention refinement" from GAP in section 2.4. Hyperparameter $\lambda \in (0, 1)$ decides how much attention priors contribute per word and region. The architecture including the implementing process can be seen in Figure 3. Finally, the answer tokens $\mathbf{y}$ are predicted using the question summary vector $\mathbf{q}$ and the output of the GANG model $\mathbf{h}'$.

$$\mathbf{h}' = \lambda \cdot \mathbf{p} \ + (1 - \lambda) \cdot \mathbf{h}$$
$$\mathbf{y} = \text{Softmax}(\text{MLP}(\mathbf{h}', \mathbf{q}))$$

where $\mathbf{p}$ is attention prior and $\mathbf{h}$ is result of GAT.

4. Experiments

We evaluate our approaches (DGA, GANG) on the same experimental conditions of baseline (GraphVQA) and our model. Because our resources were limited, the Baseline model was first tested on limited resources and the results were recorded. Since then, our model is tested with limited resources, that is, under the same conditions, and the results are recorded and compared with the two models. Of the entire GQA dataset, only 32000 questions were randomly extracted and trained. Random seeds were fixed to prevent randomness from affecting the final performance. All experiments were conducted based on 30 epochs. Table 1 shows evaluation results on GQA. All numbers are in percentages except for distribution. The lower the better for distribution.

We note that dilation-based GCNs performed better than GAT on the Validity metric with baseline architecture. Dilation-based GCNs exhibit a slower propagation of information as the distance between nodes increases. This enables the embeddings to gain a better understanding of the graph's overall information by taking into account its underlying structure. We assume this result is due to the fact that the Validity metric shows a high number if only the structure of the question can be determined.

GAT-based scene graph encoding shows the best performance for several metrics our final model (GANG) and the baseline (GraphVQA). And our final model outperforms the baseline model and interim model (DGA) by a large margin in the Accuracy metric. Three experiments were conducted on λ, the ratio of attention prior, and it was observed that performance was greatly degraded when the ratio of prior was too large.

5. Conclusion

Summary. We introduce the Directly Guiding approach, which involves concatenating the embedding vector of a specific object in the question to each node instead of directly concatenating the instruction vector. Based on the described methods, we have developed the Guided Attention Network in Graphs (GANG) model by averaging the results of the transformer prior decoder and the results obtained from the GAT. We also experimented with various scene graph encoding methods, including GCN-based, Dilation GCN-based, and GAT-based models. The GAT-based encoding demonstrated the highest accuracy among these methods, making it the chosen model for final performance boosting.

Future works. Learning the ratio through the gating function without setting the ratio of the attention prior to the hyperparameter remains a future work.

References

[1] A. Agrawal, "VQA: Visual Question Answering," arXiv.org, May 03, 2015. https://arxiv.org/abs/1505.00468

[2] P. Anderson, "SPICE: Semantic Propositional Image Caption Evaluation," arXiv.org, Jul. 29, 2016. https://arxiv.org/abs/1607.08822

[3] D. A. Hudson, "GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering," arXiv.org, Feb. 25, 2019. https://arxiv.org/abs/1902.09506

[4] D. Huang, "Finding 'It': Weakly-Supervised Reference-Aware Visual Grounding in Instructional Videos," IEEE Conference Publication — IEEE Xplore, Jun. 01, 2018. https://ieeexplore.ieee.org/document/8578721

[5] J. Johnson, "Image Generation from Scene Graphs," arXiv.org, Apr. 04, 2018. https://arxiv.org/abs/1804.01622

[6] J. Johnson, "Image retrieval using scene graphs," IEEE Conference Publication — IEEE Xplore, Jun. 01, 2015. https://ieeexplore.ieee.org/document/7298990

[7] T. N. Kipf, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv.org, Sep. 09, 2016. https://arxiv.org/abs/1609.02907

[8] T. M. Le, "Guiding Visual Question Answering with Attention Priors," arXiv.org, May 25, 2022. https://arxiv.org/abs/2205.12616

[9] G. Li, "DeepGCNs: Can GCNs Go as Deep as CNNs?," arXiv.org, Apr. 07, 2019. https://arxiv.org/abs/1904.03751

[10] R. Li, "Situation Recognition with Graph Neural Networks," arXiv.org, Aug. 14, 2017. https://arxiv.org/abs/1708.04320

[11] W. Liang, Y. Jiang, and Z. Liu, "GraghVQA: Language-Guided Graph Neural Networks for Graph-based Visual Question Answering," in Proceedings of the Third Workshop on Multimodal Artificial Intelligence, Jun. 2021, pp. 79–86. doi: 10.18653/v1/2021.maiworkshop-1.12.

[12] R. R. Selvaraju, "Taking a HINT: Leveraging Explanations to Make Vision and Language Models More Grounded," arXiv.org, Feb. 11, 2019. https://arxiv.org/abs/1902.03751

[13] P. Veličković, "Graph Attention Networks," arXiv.org, Oct. 30, 2017. https://arxiv.org/abs/1710.10903

[14] J. Wu, "Self-Critical Reasoning for Robust Visual Question Answering," arXiv.org, May 24, 2019. https://arxiv.org/abs/1905.09998