Real World Graphical Reasoning and Compositional Question Answering

Henry Friedlander, Preston Ng
Department of Computer Science
Stanford University
{hnf035,plng}@stanford.edu

Abstract

There has been much progress in recent years in the field of visual question answering. Many models are running at super-human performance, and progress appears promising. The MAC network's compositionality-inspired architecture [1] has achieved a startling 98.9% accuracy on the CLEVR dataset, which should indicate that this field is solved. However, recent papers have been shown that these 'superhuman' models are not learning as much as we, human operators, would like. In the case of the MAC network on the CLEVR dataset, MAC is simply compositionality on CLEVR's very small answer space. In this paper, we propose modifications to the MAC network by utilizing scene graphs to avoid this overfitting so that it can work in a more general setting. We chose the versatile and real world GQA dataset [2] as a test set to benchmark our results. In this report, to emphasize Natural Language aspects of machine reasoning, we decided to focus on operating over the scene graph data of GQA rather than images.

1 Introduction

From concrete concepts like language to more abstract concepts like human reasoning itself, compositionality plays a significant part in the theater of the intelligent man. However, articulating the precise qualities of compositionalities can be elusive even for humans–much less machines. Nevertheless, Gottlob Frege, the father of analytic philosophy, begins to do the concept justice. He states "I do not begin with concepts and put them together to form a thought or judgment; I come by the parts of a thought by analyzing the thought." His observations hint at the essence of compositionality and human creativity.

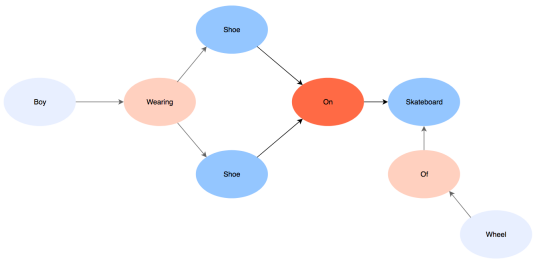
Beginning with the concept of a idea or in our case, a question, we come to some understanding with which we begin our thoughts. Over time, we can descend from our abstractions and attend to concrete parts of our thoughts. More concretely, we internalize the largest, most complex structures to understand the big picture. Then, we decompose the information, at discrete steps, to towards less complex concepts, which we articulate in an answer. This leads us to believe that compositionality and relational abstractions are linked in nontrivial ways that deep learning is apt to uncover.

This lead us to using the MAC architecture. This architecture emphasizes decomposition of questions through discrete attention steps to internalize the reasoning process. However, previous work involving the MAC architecture has focused on overloading its capacity for compositionality. Despite achieving 98.9% accuracy on the CLEVR dataset, there is much progress to be made in the area of extending the model to question-answering spaces with complex of compositionalities *as well as* complex answer spaces. These insights inspire this project.

Our contributions involve extending the MAC framework to operate over scene graphs rather over the raw images. We have decided to use the scene graphs of the GQA dataset.

Figure 1: An example of data in the GQA dataset.

(a) Original image



(b) Nodes pertaining to question

Q: What is on the skateboard in the bottom of the photo?
A: Shoe

Figure 1 is an example of an entry in GQA (only data relevant to answering the particular question is displayed). Humans follow the process Frege outlined above. They first internalize the image. Second, they would focus on the most distinctive feature of the image–the boy. Then, they notice the relationships of the image. We are modeling this process as the mental creation of a "scene graph".

Only after this, do they attend to the question and their knowledge of the scene relations informs their attentive steps over the question. Having internalized the question, they focus on the concept of the "skateboard" and retrieve the corresponding node in the knowledge graph. Finally, they traverse across the graph acording the semantic structure of the sentence finding the answer to be "show". This reasoning process is the inspiration for our model.

2 Related Work

Much research has emerged recently in using graphs as inputs to models. A couple of particularly complex network that has emerged including the graph convolutional network [4], the graph attention network[7], and the gated graph sequence neural networks [9]. They make use of the information propagation throughout the network in order to provide a better representation of the input. However, rather than adding large networks to the already complex MAC architecture, for this paper, we have chosen to create more lightweight additions.

Of course, this research is inspired by the original MAC architecture [1] that has shown to perform machine reasoning well on the CLEVR dataset [3], but does not perform well on real world images via the GQA dataset [2]. A large influence for real world visual reasoning comes from the creation of Visual Genome [5], a dataset that has enabled modeling of interactions and relationships between objects in an image. The annotations within the Visual Genome dataset is currently the largest dataset including image descriptions, objects, attributes, relationships, and question answer pairs. In fact, the GQA dataset is heavily based on the Visual Genome dataset. There are currently no baseline papers that have tried to perform machine reasoning on the GQA dataset because the GQA dataset is newer.

3 Approach

In order to extend the MAC architecture to the GQA dataset, there are various changes that are being applied, as described in more detailed below. First, the base MAC architecture is described. Afterwards, modifications to the MAC architecture is broken down into the control unit and memory unit.

3.1 Base MAC Architecture

Currently, the base MAC architecture is comprised of a recurrent network of MAC cells, each containing a control and memory (read/write) unit.

The inputs: The inputs to the model are comprised of a question and an image. The question is converted to contextual words and a question representation via a bidirectional LSTM, where the question is transformed into a positional-aware vector q_i, representing the aspects of the question relevant to the i^{th} reasoning step. The image is then converted to a knowledge base corresponding to a $14 \times 14 \times d$ image region representation tensor.

Figure 2: MAC Cell

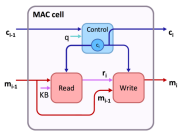
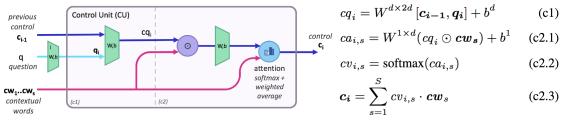


Figure 3: Control Unit



$$cq_i = W^{d\times 2d}[\boldsymbol{c_{i-1}}, \boldsymbol{q_i}] + b^d \quad \text{(c1)}$$

$$ca_{i,s} = W^{1\times d}(cq_i \odot \boldsymbol{cw_s}) + b^1 \quad \text{(c2.1)}$$

$$cv_{i,s} = \text{softmax}(ca_{i,s}) \quad \text{(c2.2)}$$

$$\boldsymbol{c_i} = \sum_{s=1}^{S} cv_{i,s} \cdot \boldsymbol{cw_s} \quad \text{(c2.3)}$$

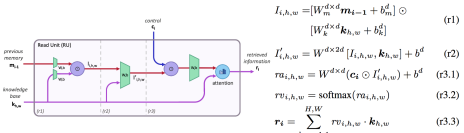
The control unit c_i: The control unit is responsible for the reasoning operation at timestep i, selectively focusing on some aspect of the question. This is represented by a soft attention-based weighted average of the question words.

Breakdown of the **control unit**:

1. Utilize a linear transformation over the concatenation of the previous control with current q_i (question $\rightarrow$ position-aware vector) (cq_i) to incorporate the previous control state.
2. Utilize a linear transformation over the output cq_i elementwise multiplied by each question word, and put through a softmax to yield an attention distribution, and finally sum the words according to this distribution in order to produce the reasoning operation c_i represented in terms of the question words.

The memory unit m_i: The memory unit holds intermediate results obtained from the reasoning process up to timestep i computed recurrently by integrating preceding hidden state m_{i-1} with new information r_i from the image, performing reasoning operation c_i. The memory unit is made up of 2 parts: a read and a write unit described briefly below.

Figure 4: Read Unit



$$I_{i,h,w} = [W_m^{d\times d}\boldsymbol{m_{i-1}} + b_m^d] \odot [W_k^{d\times d}\boldsymbol{k_{h,w}} + b_k^d] \quad \text{(r1)}$$

$$I'_{i,h,w} = W^{d\times 2d}[I_{i,h,w}, \boldsymbol{k_{h,w}}] + b^d \quad \text{(r2)}$$

$$ra_{i,h,w} = W^{d\times d}(\boldsymbol{c_i} \odot I'_{i,h,w}) + b^d \quad \text{(r3.1)}$$

$$rv_{i,h,w} = \text{softmax}(ra_{i,h,w}) \quad \text{(r3.2)}$$

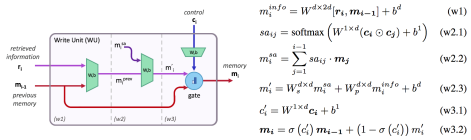
$$\boldsymbol{r_i} = \sum_{h,w=1,1}^{H,W} rv_{i,h,w} \cdot \boldsymbol{k_{h,w}} \quad \text{(r3.3)}$$

Breakdown of the **read unit**:

1. The direct interaction between the knowledge-base element $k_{h,w}$ and previous memory m_{i-1} is computed by utilizing a linear layer for both $k_{h,w}$ and m_{i-1}, and then element-wise multiplying them together in order to perform transitive reasoning by considering current important information with regards to the previous computation steps, yielding $I_{i,h,w}$.
2. Utilize a linear layer on the concatenation of $I_{i,h,w}$ and $k_{h,w}$ to incorporate new information related to the current reasoning step, yielding $I'_{i,h,w}$.
3. Retrieve relevant information based on the control unit c_i and $I'_{i,h,w}$ via element-wise multiplication, and then running that result through a linear transformation. In addition, utilize a softmax over that result producing another attention distribution over the knowledge base elements, and utilize this attention distribution to compute a weighted average, yielding r_i.

Breakdown of the **write unit**:

Figure 5: Write Unit



$$m_i^{info} = W^{d\times 2d}[r_i, m_{i-1}] + b^d \quad \text{(w1)}$$
$$sa_{ij} = \text{softmax}\left(W^{1\times d}(c_i \odot c_j) + b^1\right) \quad \text{(w2.1)}$$
$$m_i^{sa} = \sum_{j=1}^{i-1} sa_{ij} \cdot m_j \quad \text{(w2.2)}$$
$$m_i' = W_s^{d\times d} m_i^{sa} + W_p^{d\times d} m_i^{info} + b^d \quad \text{(w2.3)}$$
$$c_i' = W^{1\times d} c_i + b^1 \quad \text{(w3.1)}$$
$$m_i = \sigma(c_i')\, m_{i-1} + (1 - \sigma(c_i'))\, m_i' \quad \text{(w3.2)}$$

1. Concatenate r_i with the previous memory state m_{i-1} and utilize a linear transformation to integrate new information into the current memory state m_i.
2. Utilize self-attention via a softmax to derive an attention distribution over the previous reasoning steps, which provides information to the importance of previous timesteps with regards to the current one.
3. Finally, a memory gate is added in order to allow the model to dynamically adjust the reasoning process between the previous memory state m_{i-1} and the current one m_i', conditioned on c_i. The memory gate allows the cell to skip a reasoning step if necessary, and can pass previous memory values forward, producing the final state of m_i.

3.2 MAC Baselines with Images

CLEVR Dataset baseline: Since this dataset is simpler, the MAC network is able to perform well, achieving an overall accuracy of 98.9%.

GQA Dataset baselines: Afterwards, the MAC network was applied to the GQA dataset in order to gather a baseline without any modifications. The GQA dataset was assessed with the same base MAC network that performed well on the CLEVR dataset. Currently, there is a validation dataset accuracy of 61.91%, a large drop compared to the CLEVR dataset.

3.3 Finished Modifications

The control unit: In the base MAC architecture, the control unit attends to some part of the question utilizing a linear transformation that takes into account the overall question and the preceding reasoning operation. Instead of utilizing a linear transformation, we coded and incorporated an LSTM to incorporate control information from all previous timesteps. This lead to an improvement in the CLEVR dataset by 0.2%, but minimal improvements if any in the GQA dataset.

The input: GQA encodes an image as a scene graph containing an image's objects, attributes, and relations. This encoding is different than the current image input, which utilizes a $14 \times 14 \times d$ tensor that pertain to various areas of an image. Instead, modifications to extend the MAC architecture to the GQA dataset includes changing the image input to a node emphasized relationship embedding.

We propose an altered knowledge graph to vector encoding scheme that stresses the nodes relevant to the compositional structure of MAC. Instead of encoding each image as various area representations, we utilize a knowledge graph representation where each object k_{obj_i} is represented as a node.

$$k_{obj_i} = [a, attributes(a), r_1 \rightarrow b, \cdots],$$

We will now describe how to obtain this *node emphasized relationship embedding*. We lookup a node emphasized relationship embedding for each node representation k_{obj_i}.

$$emb(k_{obj_i}) \in \mathbb{R}^{n\times d},$$

where n is the number of representation values for a particular object node. The embedding is taken from an unfixed GloVe embedding so that the embeddings can learn to capture the nuances of the scene graph relations as training progresses.

There were two approaches taken to attend to the image nodes:

1. **Flatten:** The first naively flattens the object and node representation dimensions so that the final output is an attention map over $\mathbb{R}^{mn}$ values, where m represents the number of nodes and n represents the number of representation values for a node. There is an issue with this approach though because this reverses the emphasis on nodes, and instead broadly represents nodes as a collection of node representation items.

2. **Average Pool**: The second approach compresses the node representation items down utilizing an Average Pooling approach on all the embeddings of a particular node representation $emb(k_{obj_i})$. Average pooling reduces the size of $k_{obj_i} \in \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{d}$. Utilizing this approach allows the model to attend to particular node representations correctly. Object nodes are now the emphasis of image representation rather than various image regions. We have now converted $K_{h,w} \in \mathbb{R}^{h \times w \times d} \rightarrow K_i \in \mathbb{R}^{m \times d}$, where m represents the number of object nodes in a particular image. This approach was the contributing factor in the large increase in accuracy for the MAC network on the GQA dataset.

After objects are encoded, we utilize a GRU to attend over the whole scene graph based on training.

3.4 Current Modifications

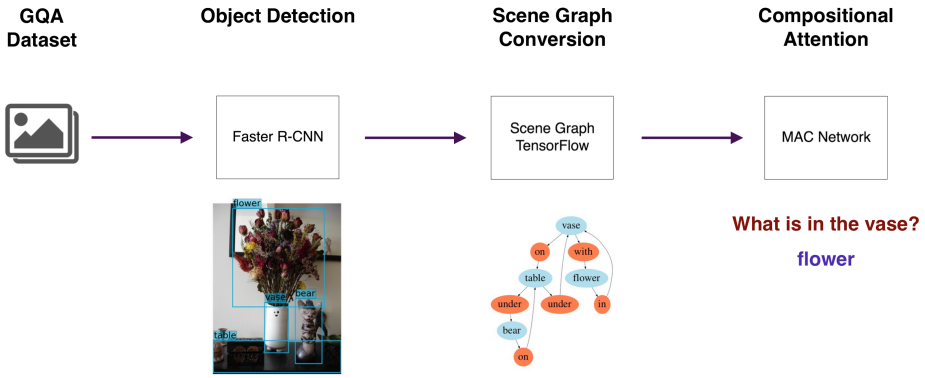
The input: In order to enable GQA to train end to end from image to output (rather than scene graph to output), image information needs to be encoded to a scene graph that can be passed into the current MAC network. We propose a 3 stage process:

1. **Object Detection:** Faster R-CNN [6] is utilized in order to perform object detection. This is a necessary step in order for scene-graph-TF [8] to understand object nodes within an image.
2. **Scene Graph Creation:** Scene graphs can be created via scene-graph-TF [8], which takes in object detection information from Faster R-CNN (step 1), and generates scene graphs via iterative message passing.
3. **MAC Network**: Once scene graphs are generated, the current updated Mac Network can be utilized to generate question answering on new images.

Preprocessing progress has been made in order to create a new dataset that can be used with Faster R-CNN for object detection, so that scene-graph-TF can generate scene graphs that can be feed into the current MAC network.

A visualization of the whole process is as follows:

Figure 6: Newly Proposed GQA End to End Pipeline



4 Experiments

4.1 Data

CLEVR: The base MAC network [1] utilizes the CLEVR dataset [3] for the original visual reasoning challenge. This dataset contains a smaller and finite answer space on which the MAC network performs well as the state-of-the-art for this particular dataset. The CLEVR dataset is used as a standard for the possibility of how well the MAC network can perform.

GQA: The GQA dataset [2] contains real world data containing information about compositional questions over real-world images. Images are associated with a scene graph of the image's objects,

attributes, and relations. In addition, semantic representations are utilized for both scenes and questions to address language priors and conditional biases. Questions within the GQA question set involve multiple reasoning skills, spatial understanding, and multi-step inference which prove more difficult than other datasets utilized previously. This dataset holds 74939 training scene graphs, as well as 10696 validation scene graphs.

4.2 Evaluation Method

The GQA system offers a large selection evaluation metrics including: accuracy, consistency, validity, plausibility, distribution, and grounding scores. These evaluation metrics will be used to evaluate the system as a whole.

4.3 Experiment Details

Baseline models:

- Epochs: 25
- Learning Rate: 0.0001
- Batch Size: 64
- Dropout:
  - encoder input (dropout of the rnn inputs to the Question Input Unit): 0.85
  - encoder state (dropout of the rnn states of the Question Input Unit): 1.0
  - stem (dropout of the Image Input Unit (the stem)): 0.82
  - question (dropout on the question vector): 0.92
  - memory (dropout on the recurrent memory): 0.85
  - read (dropout of the read unit): 0.85
  - write (dropout of the write unit): 1.0
  - output (dropout of the output unit): 0.85
- Train time: 1 day
- GPUs: 2 TITAN Xp
- CLEVR dataset net length (network length (number of cells)): 16
- GQA dataset net length (network length (number of cells)): 4

4.4 Results

Results via accuracies for both the CLEVR and GQA datasets are shown below. There are currently no test datasets for GQA, so only training and validation accuracies are included in the results.

Table 1: Accuracies for the image-based Datasets

Model	CLEVR(Test)	GQA (Val)
MAC Base Architecture	98.9	61.91

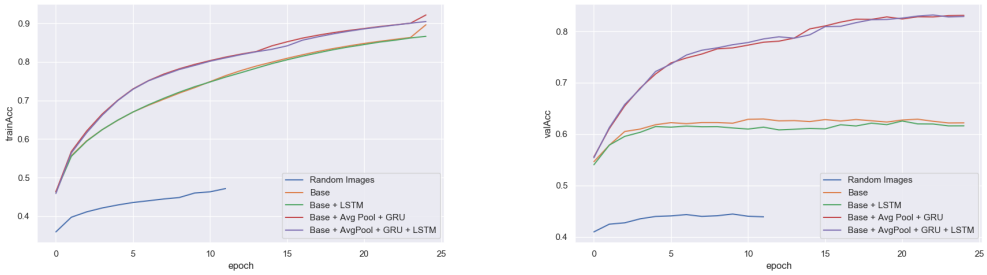
Table 2: Scene Graph Accuracies for MAC + SG Base Model on GQA Dataset

Model (MAC + SG)	Train Accuracy	Val Accuracy
Random Images	47.17%	43.90%
Base + Flatten	85.34%	62.90%
Base + Flatten + LSTM	83.13%	62.13%
Base + Avg Pool + GRU	92.17%	83.11%
Base + Avg Pool + GRU + LSTM	**89.56%**	**83.19%**

4.5 Quantitative Evaluation

In table 1, the high accuracies for the CLEVR dataset is expected because it is a simpler dataset with a finite number of answers. GQA accuracies are expected as well because it is a more complex dataset that includes real world images (sidenote: only validation accuracies are currently available due to the in progress changes to extend the MAC architecture to the GQA dataset).

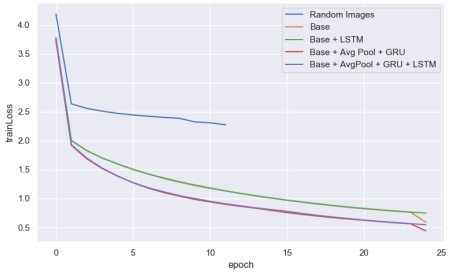
Figure 7: Accuracies



(a) Train Accuracy

(b) Validation Accuracy

Figure 8: Train Loss



Our best performing model was the Base + Avg Pool + GRU + LSTM. This high performance with this model aligns with our hypothesis that subdividing the knowledge base up into compression steps using the GRU final states, context awareness using Avg Pooling layers, and increased control robustness creates the superior model.

Converting the MAC network to operate on scene graphs enabled the system to have a 21.28% increase in validation accuracy (83.19%) over the previous model that operated on images (61.91%). This is a major step towards enabling a better VQA system as a whole on datasets that have real world images, but is only part of the process. Because we are operating on fully observed scene graphs, it is expected to have a large increase in accuracies.

5 Analysis (Qualitative Evaluation)

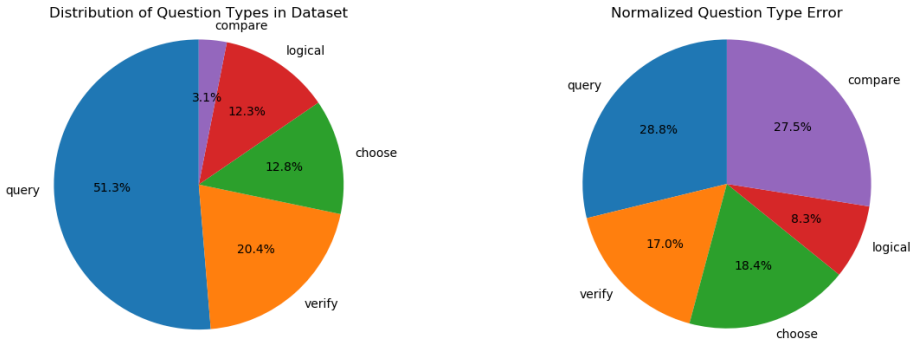
The training dataset (74939 scene graphs) was used for training the MAC network. The validation dataset (10696 scene graphs) is ran through this network in order to gather quantitative results (in the Results section), as well as qualitative results (shown below).

In the above pie charts, we show the distributions of various question types (*query*, *verify*, *choose*, *logical*, *compare*) and their normalized error rates. There is an overwhelmingly larger percentage of *query* questions (e.g. What is in the vase?) which require a free form response. In addition, we find that *query* type of questions have the largest error rate (normalized). This seems intuitive because free form responses are harder to elicit than logical (yes/no type questions).

In addition, there are some query questions that seem very difficult to answer due to the way the scene graphs are being created.

In figure 10, the true answer is **blue** because that is what we as humans perceive, but the system that created the scene graphs characterized the curtains as being green. There are other variables to consider now since the MAC network is a submodule of a larger system. It is important to look at how individual parts of a system contribute to the final outcome. Here, it is clear that there are other upstream parts of the system that need to be analyzed in depth so that progress can be made to make the system more robust. Since this bottleneck is part of an upstream submodule that creates the scene

Figure 9: Distibutions based on validation dataset (10696 scene graphs, 10000 questions)



(a) Distribution of Question Types

(b) Distribution of errors (Normalized)

Figure 10: Wrong answer due to upstream scene graph creation.

Q: What color are the curtains?
Pred: Green, **A:** Blue

graphs, it is plausible that the MAC network submodule is more accurate and robust than is shown from the quantitative evaluation due to these inconsistencies.

6 Conclusion

We have introduced changes to the Memory, Attention, and Composition (MAC) network that now operates on scene graphs in order to perform machine reasoning. By operating on scene graphs, the MAC network can now perform graphical reasoning on real world examples. We have replaced the image input in the read unit to encoded knowledge graphs for a particular image. By replacing images with node emphasized embeddings, individual nodes are attended to rather than image regions. Limitations to our work include not having a full end to end system, but rather a submodule to a whole system for visual reasoning. For future work, we want to enable the system to work on new image input by implementing an image to scene graph system, and then passing the scene graph input to the current MAC network that operates on scene graphs. By doing so, we will complete an end to end model that can perform real world visual reasoning and compositional question answering.

Mentor: Sahil (CS224n) Drew Hudson (External)

References

[1] Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning.

[2] Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for compositional question answering over real-world images. *Conference on Computer Vision and Pattern Recognition (CVPR)* .

[3] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

[4] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *CoRR* abs/1609.02907. http://arxiv.org/abs/1609.02907.

[5] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. https://arxiv.org/abs/1602.07332.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*.

[7] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. *International Conference on Learning Representations* Accepted as poster. https://openreview.net/forum?id=rJXMpikCZ.

[8] Danfei Xu, Yuke Zhu, Christopher Choy, and Li Fei-Fei. 2017. Scene graph generation by iterative message passing. In *Computer Vision and Pattern Recognition (CVPR)*.

[9] Daniel Tarlow Marc Brockschmidt Richard Zemel Yujia Li. 2016. Gated graph sequence neural networks. https://arxiv.org/abs/1602.07332.