

Object-based reasoning in VQA

Mikyias T. Desta
IBM Research, Almaden
mtdesta@us.ibm.com

Larry Chen*
University of Chicago
larrychen@uchicago.edu

Tomasz Kornuta
IBM Research, Almaden
tkornut@us.ibm.com

Abstract

Visual Question Answering (VQA) is a novel problem domain where multi-modal inputs must be processed in order to solve the task given in the form of a natural language. As the solutions inherently require to combine visual and natural language processing with abstract reasoning, the problem is considered as AI-complete. Recent advances indicate that using high-level, abstract facts extracted from the inputs might facilitate reasoning. Following that direction we decided to develop a solution combining state-of-the-art object detection and reasoning modules. The results, achieved on the well-balanced CLEVR dataset, confirm the promises and show significant, few percent improvements of accuracy on the complex "counting" task.

1. Introduction

Due to successful application of machine learning, and deep learning [24] in particular, we have lately observed great progress in traditional Computer Vision (CV) tasks such as image classification [23], object detection and image segmentation [10]. These advances have pushed researchers towards more complex tasks such as image caption generation [21] and VQA [4, 28]. Both of those tasks combine Computer Vision with Natural Language Processing (NLP) and high-level reasoning, thus require utilization of multi-modal knowledge beyond a single sub-domain (such as CV). In the case of caption generation, however, there may be several valid captions describing a given image. Thus, it is hard to define a quantitative evaluation metric to track the progress. In contrast, in VQA the image and question pairs might come with the ground-truth answers, enabling to measure the system accuracy. Moreover, as there might be many questions referring to the same image, the system is in fact forced to learn various reasoning types depending on the type of question. For those reasons VQA is considered to be AI complete [4] and renews the hope of building machines that could pass the Turing test in

open domains [28].

During the last three years several complex VQA datasets have been introduced, however, many of the solutions show only marginal improvements over the strong baselines. As there are many influencing factors it is often hard to tell which modules of the system are working properly and why. The CLEVR dataset [18] was created with aim of pushing forward the progress on VQA in a more systematic manner, with the main focus put on the validation of reasoning skills. The dataset delivers a well-balanced set of images and questions, where the information associated with each image is complete and exclusive. Along with the dataset the authors provided several, sometimes quite sophisticated baselines, and concluded that those models have not learned the semantics of spatial reasoning at all.

In this paper we decided to focus on that aspect of the problem, i.e. spatial reasoning. The hypothesis we would like to validate is whether the operation on high-level and abstract facts extracted from the image might improve the accuracy of the system (as suggested e.g. in [2]). For that reason, we developed a solution operating on object-relation-object triplets [7], similar to relational network [35], however, instead of using features extracted from the images, we rely on the information associated with the objects detected in the image. The achieved results are comparable with current state-of-the-art solutions in terms of overall accuracy and show significant improvement on the counting task.

2. Related work

Research on VQA has resulted in many interesting solutions. Those, however, could not have been developed without the existence of proper datasets and metrics for evaluation and comparison of the results.

From the historical perspective, the first dataset designed as a benchmark for the VQA task was DAQUAR (DAtaset for QUestion Answering on Real-world images) [27]. The dataset contains 1.5k RGB-D images of indoor scenes from NYU-Depth v2 dataset [37], with annotated semantic segmentations and question/answer pairs of two types: synthetic questions/answers generated automatically on the ba-

*Work done during an internship at IBM Research, Almaden.

sis of NYU annotations and human question/answers collected from 5 annotators. The limitations of DAQUAR (i.e. restriction of answers to a predefined set and strong bias in human annotations) resulted in releases of several other datasets, most prominently: COCO-QA [33] (based on images from MS COCO dataset [25], where substantial effort was made in order to increase the scale of training data), VQA [4, 11] (consisting of two sets, VQA-real with natural images, and VQA-abstract with cartoon images, and providing 17 additional (incorrect) candidate answers for each question) and Visual Genome [22] (currently the largest VQA dataset, with 1.7 million question/answer pairs and with images contents additionally described by structured annotations in the form of scene graphs).

The paper which introduced DAQUAR [27] also laid the foundations for evaluation of the system accuracy, and in consequence, monitoring the overall progress in the field. The authors proposed two basic evaluation metrics. First, by simply measuring the accuracy with respect to the ground truth answer using string matching. Second, by using the Wu-Palmer Similarity (WUPS), enabling to evaluate the similarity between common objects by their distance in a taxonomy tree. Other evaluation methods include the previously mentioned one in a multiple-choice settings [4] or the "fill in the blanks" approach proposed along with the Visual Madlibs dataset [48]. As the field started to mature, researchers discovered the importance of biases in the images and questions, which triggered the release of more balanced datasets (such as VQA v2 [11] and CLEVR [18]). It also led to the analysis of the importance of the common-sense knowledge required to solve a given task, which in turn resulted in efforts towards delivering ground truth containing rich description of the scene, facilitating both training and evaluation (e.g. Visual Genome [22]).

However, the most interesting achievements in the VQA field are the novel algorithms and neural architectures. In [43] the authors proposed to distinguish four categories: joint embedding approaches, attention mechanisms, compositional models and models using external knowledge bases, which also summarizes the four main research directions in VQA.

The efforts in joint embedding focus on the methods for combining multi-modal representations. As in VQA there are two distinct input modalities (image and text), which makes this problem similar to the problems found in other multi-modal domains. For example the projection of user and item embeddings into the common representation space in neural recommender systems [15]. Exemplary approaches developed for the VQA problem domain include e.g. Multimodal Compact Bilinear pooling (MCB) [9] method that performed joint embedding of visual and text features, or Relational Networks (RN) [35] where embedded question was concatenated with features extracted from

pairs of image regions, enabling the system to reason about the relation between objects being present in those regions.

As some questions might require more than one reasoning step, a lot of researchers focused on attention mechanisms, which were initially introduced for neural translation [5], and later on adapted to the Question-Answering problem [42]. In [36] the authors introduced a simple attention model, where an embedded question was used for the generation of an attention mask (called region-question relevance), which was subsequently used for the calculation of a weighted vector of concatenated question and image features. Yang et al. [47] introduced Stacked Attention Networks, which are able to infer the answer iteratively. This architecture includes two attention layers, enabling two reasoning steps (in [42] called "hops"), each driven by a different attention mask over the image features. The influence of number of hops was analysed in [46], where the authors introduced a spatial attention-based model called Spatial Memory Network. Here, the model uses each word embedding to capture a fine-grained alignment between the image and the question. The results indicated the superiority of the two-hop model over the model with a single hop. The described methods above focused only on visual attention driven by the question. In contrast to those ideas, a major innovation was proposed in [26], where Hierarchical Co-Attention (HieCoAtt) model processed image and question symmetrically, with the image guiding the attention over the question and vice versa. In a quest to assess the quality of the attention models, in more recent works researchers have tried to validate whether the attention models learned by neural models focus on the same image regions as humans do [8].

The third direction results from the observation that different tasks might require totally different reasoning processes. For example in the "query attribute" task the system should focus its attention on a single aspect of the scene, whereas in "counting" it should perform a sequence of actions, focusing on the objects of interest one by one. This observation led to the development of Neural Module Networks [3], where semantic parsers (e.g. Stanford Parser) were used for question-driven dynamic assembly of the internal structure of the system, which is different for every type of question. Most recent advances include program-induction [19] consisting of a program generator and program executor, where the former is a sequence-to-sequence model (a pair of LSTMs, i.e. Long Short Term Memory recurrent neural nets [16]) responsible for constructing of an internal representation of the reasoning process to be performed, whereas the latter is responsible for execution of the resulting program in order to produce the answer.

The last direction includes research on models using external knowledge bases (i.e. external sources of data) for answering the question. One such solution is proposed in

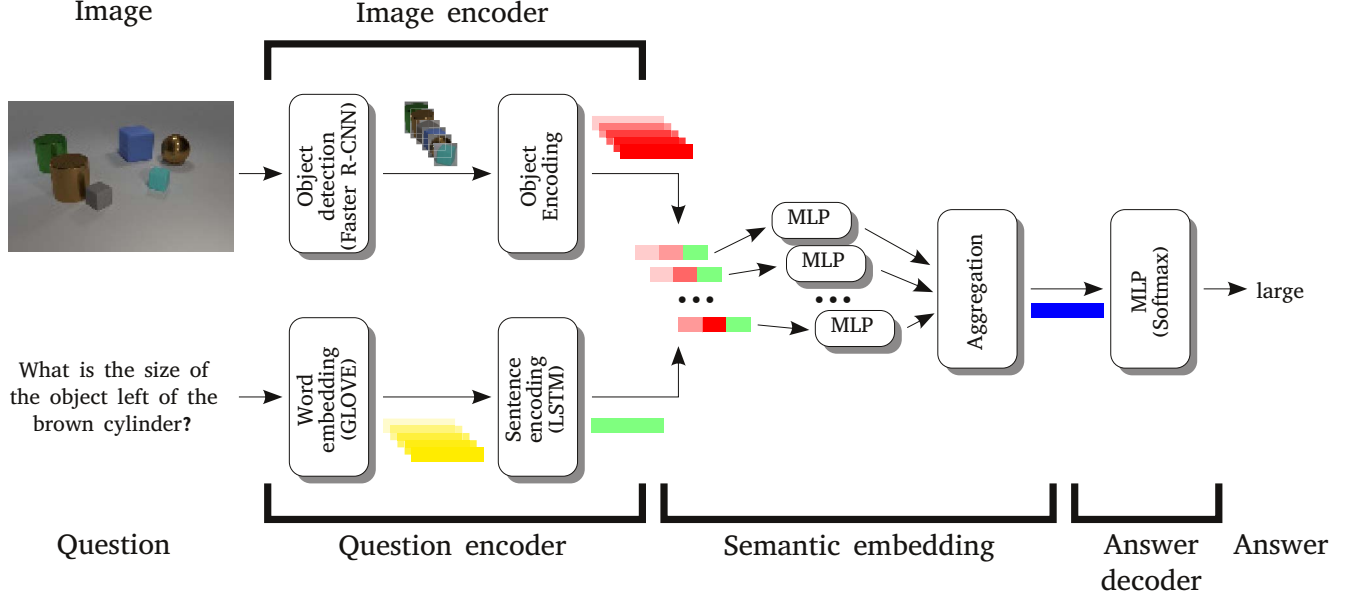


Figure 1: General architecture of the proposed system

[44], where the authors developed a framework that extracts image-related information from the DBpedia knowledge base, enabling it to answer a broad range of questions, often requiring knowledge that could not be inferred directly from the image. The improvements achieved by results e.g. in the paper proposing the FVQA dataset [40] strongly indicate that utilization of supporting-facts extracted from large-scale structured knowledge bases might be the key ingredient for solving the Visual Turing test. As there are other findings that also suggest the need for operation on high-level, abstract facts extracted from the image (e.g. [2]), we decided to investigate that direction further.

3. Our solution

Architecture of our system is presented in Fig. 1. It extends the Encoder-Decoder [6] architecture, which originally consisted of two RNNs, the first one used for encoding a sequence of input symbols into a fixed length representation, and the other for decoding that representation into another sequence of output symbols.

In VQA there are two input channels with different modalities, thus we need two types of encoders and an intermediate module responsible for projection of both the visual and language representations into a common semantic embedding space. Following that scheme, Xiao et al. [45] adopted the convolutional layers of the VGG network followed by an average pooling across the output features as the “visual” encoder and used a two-layer stacked LSTM network as the “language” encoder. The authors assumed that the output of the language encoder is already in the

semantic space and they used a two-layer-perceptron to project the output of the visual encoder into the same space.

Similarly, the Refined Ask Your Neurons architecture [29] consisted of a visual encoder, question encoder, a multimodal embedding module combining both encodings into a joint space, and an answer decoder. The authors compared results achieved for several combinations of different question (e.g. BoW, CNN, GRU and LSTM) and visual (several classical architectures including AlexNet, GoogLeNet, VGG-19, ResNet-152) encoders.

3.1. Image encoder

In our system the image encoder realizes two major operations: object detection (OD) and object encoding. The diagram for object detection is shown in Fig. 2. In a given t -th image I^t we detect a set of N objects:

$$d\mathbf{O}^t = [d\mathbf{o}_0^t, d\mathbf{o}_1^t, \dots, d\mathbf{o}_{N-1}^t], \quad (1)$$

each defined as $d\mathbf{o}_n^t = \langle d\mathbf{i}_n^t, d\mathbf{b}_n^t \rangle$, where $d\mathbf{i}_n^t$ denotes the class identifier (1 to 96) of the object and $d\mathbf{b}_n^t$ contain four parameters defining the object bounding box. We decided to use Faster R-CNN [34] with ResNet-101 [14] pre-trained on COCO dataset, since it is one of the most reliable methods for object detection currently available.

In the next step (presented in Fig. 3) we decode each object $d\mathbf{o}_n^t$ and retrieve set of its attributes:

$$a\mathbf{o}_n^t = \langle c_n^t, m_n^t, s_n^t, f_n^t, x_n^t, y_n^t \rangle, \quad (2)$$

where c_n^t is the object color, m_n^t represents material it is made from, s_n^t indicates size, f_n^t represents its shape (form)

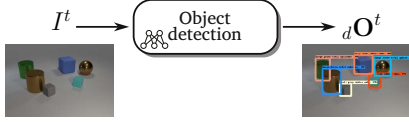


Figure 2: Dataflow diagram of object detection

and x_n^t and y_n^t describe its position in the image. Simple encodings of each of those attributes (we used one-hot in case of the former four and bucketing in the case of positional arguments, as explained in sec. 4.4) and concatenation of the resulting vectors form the encoded object description eO_n^t .

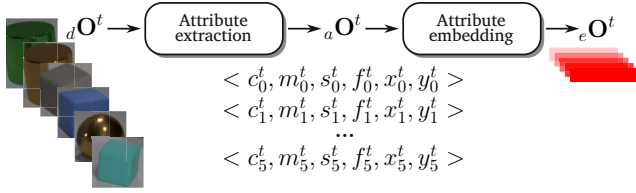


Figure 3: Dataflow diagram of object encoding

3.2. Question encoder

The diagram for question processing is shown in Fig. 4. We start with the question consisting of several words:

$$\mathbf{Q}^t = [q_0^t, q_1^t, \dots, q_{W-1}^t], \quad (3)$$

where W denotes the number of words constituting a given question. Next, we use the GloVe word embedding model [32] to encode question words: $e\mathbf{Q}^t = [eq_0^t, eq_1^t, \dots, eq_{W-1}^t]$. Finally, we pass the encoded words one by one as inputs to the LSTM [16] to produce a list of encoded output:

$$s\mathbf{Q}^t = [sq_0^t, sq_1^t, \dots, sq_{W-1}^t]. \quad (4)$$

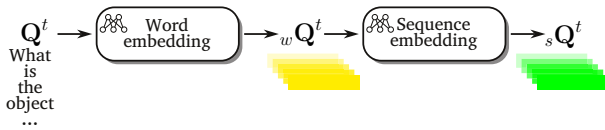


Figure 4: Dataflow diagram of question encoding

3.3. Semantic embedding

There are many ways for combining image and question features [38], such as concatenation, element-wise product or bilinear operation. In our solution, similar to the relational network module reported in [35], we form a set of feature vectors by concatenating pairs of vectors representing two objects with the encoded question, i.e. the last output of LSTM (sQ_{W-1}^t). This approach is somehow similar to object-object-relation triplets used in [7].

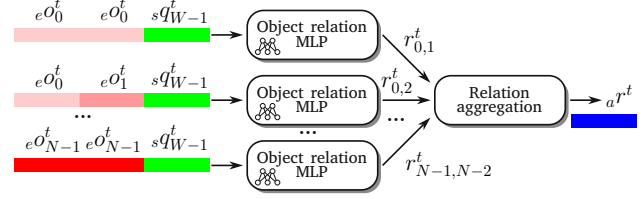


Figure 5: Dataflow diagram of semantic embedding

Then, individual triplets are passed through a four layer MLP (Multi-Layer Perceptron), each with 512 units and ReLU non-linearities, which results in a vector of relations:

$$\mathbf{R}^t = [r_{0,1}^t, r_{0,1}^t, \dots, r_{N-1,N-1}^t]. \quad (5)$$

Those relations are further aggregated into ar^t . We investigated several aggregation methods, but simple summation appeared to give the best results – please refer to sec. 4.3.

3.4. Answer decoder

Finally, we cast the answering problem as a classification task and pass the aggregated relation ar^t through three MLP layers consisting of 512, 1024, and 29 units. A 2% dropout layer was added before the last MLP layer. ReLU non-linearities were added in each layer except the last one, where we applied the softmax function. The result from the softmax was finally decoded using the word dictionary used for initial word embeddings in Question encoder (Fig. 6).

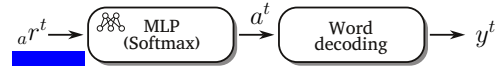


Figure 6: Dataflow diagram of answer decoding

4. Analysis of the results

4.1. Question encoding

We have pretrained the GloVe word embedding model on the Google News corpus [30] (3 billion running words) word vector model and then generated embeddings for the dictionary formed on the basis of 93 words present in CLEVR questions and answers.

4.2. Image encoding

Utilization of object detector required its prior training on images with ground truth bounding boxes and object classes. Since scene descriptions in the CLEVR dataset do not provide bounding boxes, we wrote a program for their generation. We calculated the bounding boxes for every object present in the scene relying on the associated meta-data from the scene description, i.e. using object position

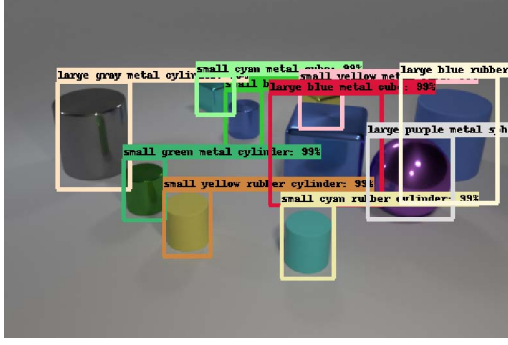


Figure 7: Exemplary CLEVR scene with objects detected with our trained object detector

in the image (2d pixel coordinates and rotation), its position in the Cartesian space (3d coordinates) along with its size and shape. In addition, we have assigned each object a class identifier taking into account the possible combinations of object attributes as defined in (2), which resulted in 96 unique classes.

An exemplary scene is presented in Fig. 7. We compared the reconstructed scene from the OD and the ground truth scene from the scene description to check how accurately the OD network is predicting the scenes. We were able to get a precision of 0.99 and a recall of 0.99, thus we find the results satisfactory.

4.3. Comparison with CLEVR baselines

We have reproduced the majority of baselines from the original CLEVR paper [18], achieving similar results. In our implementations we used the TensorFlow framework [1] and the associated object detection API [17]. As CLEVR does not provide ground truth for the test set, we are reporting the results achieved on the validation set. During training we have used a learning rate of $1.0e^{-4}$. In the last step of Semantic embedding we have tried several aggregation methods, including simple concatenation of acti-

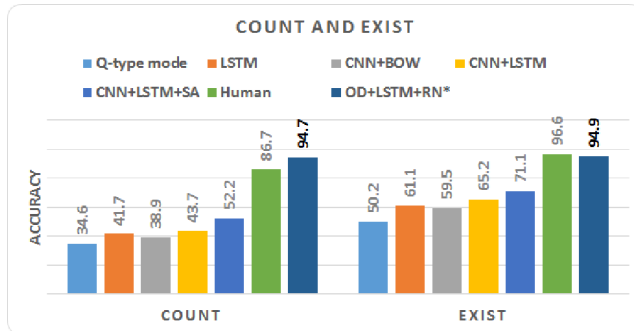


Figure 8: The comparison of accuracy of our solution in Count and Exist tasks with the CLEVR baselines

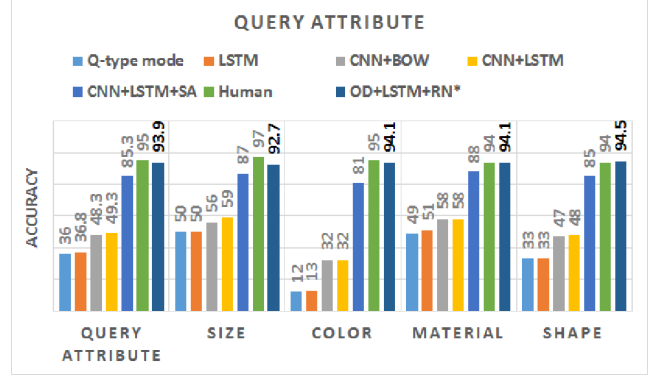


Figure 9: The comparison of accuracy of our solution with the CLEVR baselines in Query Attribute task

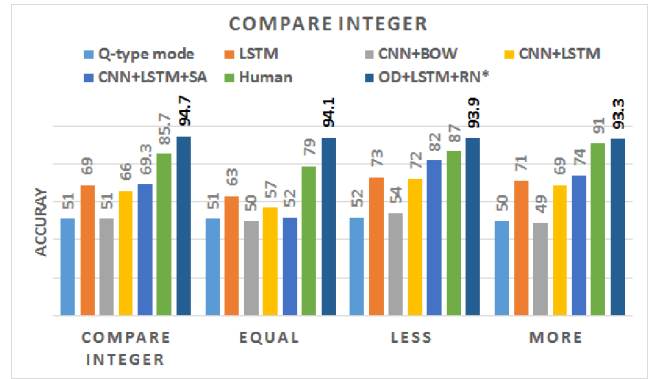


Figure 10: The comparison of accuracy of our solution with the CLEVR baselines in Compare Integer task

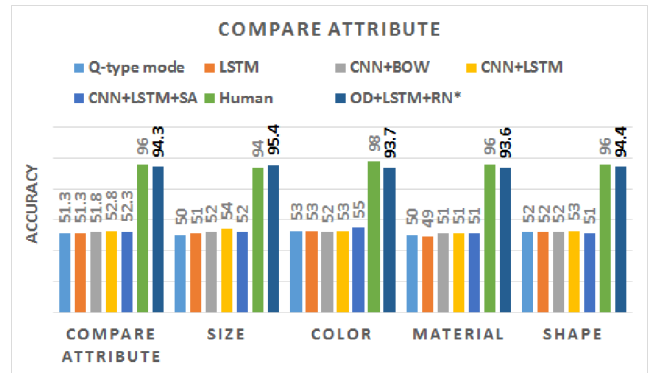
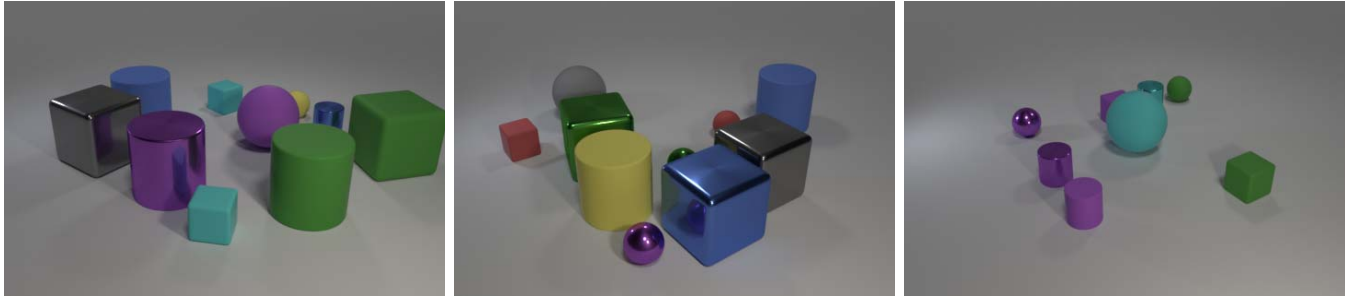


Figure 11: The comparison of accuracy of our solution with the CLEVR baselines in Compare Attribute task

uations of Object relation MLPs, using mean values of those activations and their sum. The experiments have shown that summation of those activations gives the best results (accuracy of 94.6% in comparison to e.g. 92.9% in the case of using mean value). The simple explanation is that summation ensures the invariance with respect to the order of



- (a) **Question:** what number of objects are small cyan objects that are in front of the gray object or big things that are in front of the big rubber block ?
Predicted Answer: 3
Ground Truth: 3
- (b) **Question:** how many balls are either big blue matte objects or small red things ?
Predicted Answer: 1
Ground Truth: 1
- (c) **Question:** what number of things are either small shiny objects that are behind the purple metallic cylinder or tiny purple metallic balls ?
Predicted Answer: 2
Ground Truth: 2

Figure 12: Exemplary result on counting with heavily occluded object

objects [35].

We have analyzed the results using the same task-oriented criteria as the original paper did. The accuracy on "exist" and "count" tasks is presented in Fig. 8, where the existence questions ask whether a certain type of object is present, while the count questions ask for the number of objects fulfilling some conditions. The Query Attribute task contains questions asking about an attribute of a particular object (Fig. 9). The Compare Integer questions ask which of two object sets fulfilling given conditions is larger (Fig. 10). Results for Attribute Comparison task, where questions ask whether two objects have the same value for a given attribute, are presented in Fig. 11.

Finally, we present the comparison of the overall accuracy of our solution against all the CLEVR baselines in Fig. 13. Our solution has shown major improvement over all the baselines, including the human accuracy collected using Mechanical Turk.

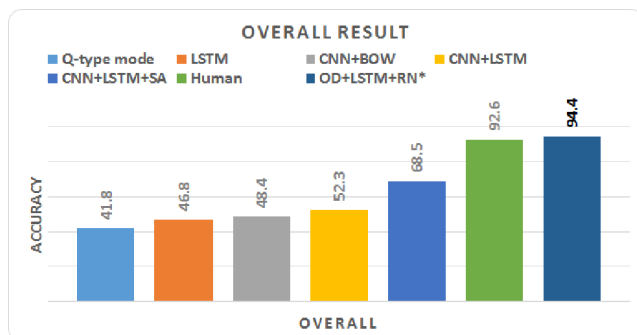


Figure 13: The comparison of overall accuracy of our solution with CLEVR baselines

4.4. Comparison with results from RN

In Fig. 14 we present the comparison of our results (shown in orange) with the results reported in [35] (shown in blue). As one might notice, our overall accuracy is slightly, by around 1%, lower. However, interestingly our solution seems to give more consistent results for different tasks and achieves better results in 2 out of 5 tasks. In particular, we achieved much better accuracy (by almost 5%) in the counting task, which is currently considered to be one of the hardest tasks in VQA (e.g. [39]). In order to realize the counting task, the system is supposed to perform several quite different operations, i.e. understanding what type of objects to focus on, finding those objects in the image and finally counting the instances. Our results indicate that a prior object detection naturally facilitates counting, which we treat as support for our hypothesis that operation on more abstract facts facilitates reasoning in general. In Fig. 12 we present a few hard cases, where our system was

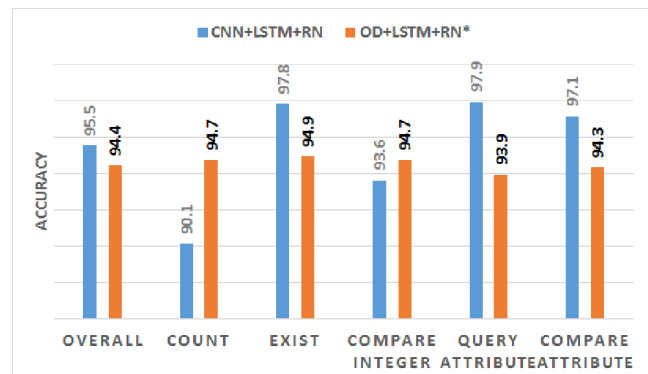


Figure 14: Comparison of our approach with results reported in the paper on Relational Networks

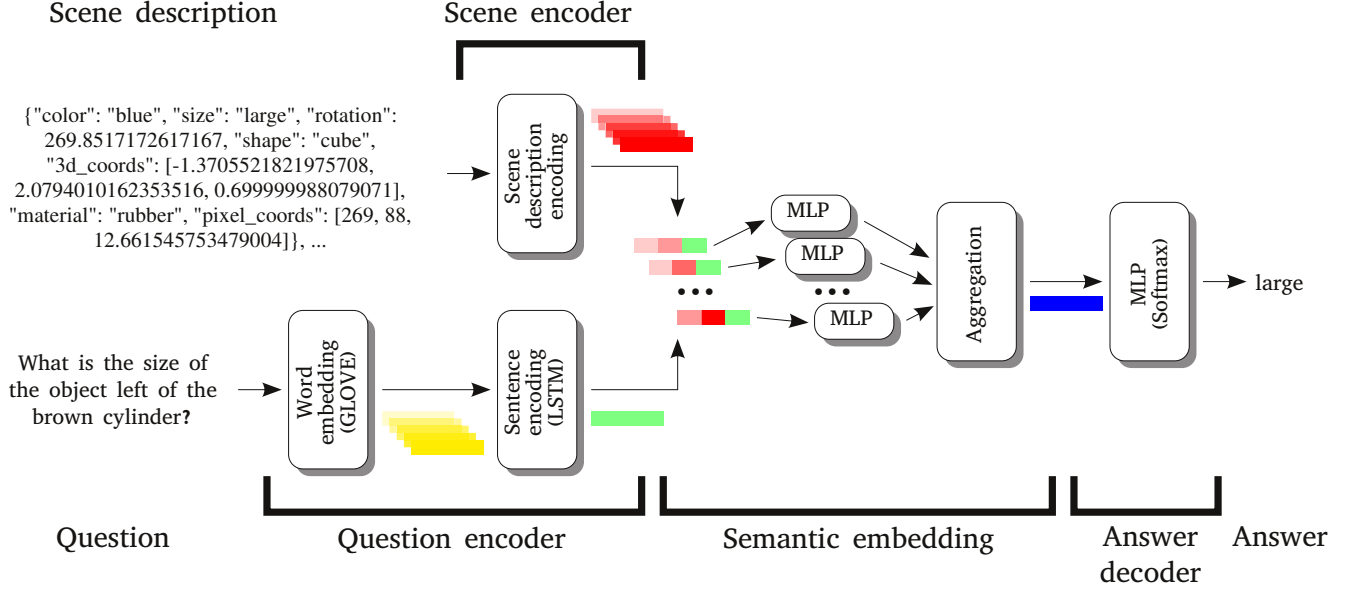


Figure 15: The architecture of the system with scene description as input used for comparison with the *CLEVR with state description* baseline from [35]

able to provide the correct answer. In particular, it managed to properly answer questions about the scenes presented in Fig. 12b and Fig. 12c despite the heavy occlusions of the objects of interest, which in [35] was suggested to be the main reason for failures in CLEVR.

We further investigated why our solution does not improve in the other three tasks. After validating the accuracy of object detection module (as reported in Sec. 4.2) we decided to reproduce the experiments from [35] in which the Image encoder was replaced by the information retrieved straight from the scene description. The architecture of the resulting system is presented in Fig. 15. Despite using the same learning hyperparameters for relational MLPs we achieved the overall accuracy of 94.5%, which is almost 2% worse from the accuracy of 96.4% reported for that setting in the paper [35]. However, as the authors did not report how they encoded the scene, we decided to further investigate that research direction. As the object attributes form small dictionaries, we assumed that simple one-hot encoding is sufficient. However, the object position expressed in pixel coordinates in the images (x_n^t and y_n^t) might be encoded in several different ways, which in turn might influence the final accuracy. Therefore, we have performed several experiments with different methods of encoding the object positions, briefly explained below.

4.4.1 One-hot encoding

In this approach we simply converted the object x_n^t and y_n^t pixel coordinates into two separate vectors using one-hot

encoding and concatenated them into one vector of length 800 (as we have 480 possible values for x_n^t and 320 for y_n^t). As can be seen from tab. 1 this method was not able to generalize as well as the others.

4.4.2 Bucketing

Here we grouped n consecutive pixels to buckets along the x and y axes separately, which resulted in a number of buckets for x and y . We did experiments on three different bucket sizes, i.e. 15, 20, and 30, which resulted in 32, 24, and 16 buckets for the x component and 22, 16, and 11 buckets for the y component respectively. When given the object position coordinate x_n^t belongs to a range of a given bucket we convert the bucket number to vector using one hot encoding. We perform the same for y_n^t and concatenate both results. It can be observed in tab. 1 that the solution generalizes better when the bucket size is chosen to be 20.

4.4.3 Object enumeration

In this method, instead of relying on the pixel coordinates we move to a more abstract representation and encoded the position of the object in relation to the positions of other objects in the scene. For x and y axes we form two separate lists, representing the order of the objects according that axis. For instance, for three objects in a scene, the very top object out of the three will have a value of one and the bottom object will have a value of three. Finally, we encoded such represented positions using one hot encoding

Encoding	Bucket size	Overall	Count	Exist	Compare numbers	Query attribute	Compare attribute
One hot	–	89.8	90.5	89.7	89.5	90.6	88.5
Bucketing	15	93.9	93.9	93.7	93.6	94.6	93.5
Bucketing	20	94.5	93.6	94.7	93.3	95.2	94.4
Bucketing	30	93.6	93.4	93.1	93.2	94.9	93.4
Enumeration	–	93.3	93.2	93.7	92.9	93.5	93.1
Results from [35]	–	96.4	–	–	–	–	–

Table 1: Results on different position encodings for solution with Image encoder replaced by the information retrieved straight from the scene description

and concatenated both encoded coordinates for every object.

Results of all the above mentioned experiments are presented in Tab. 1. The main finding is that our model was able to generalize well when we grouped twenty pixel values into one bucket. Unfortunately, these experiments did not improve the accuracy to the level reported in [35].

5. Conclusions

In this paper we have focused on reasoning using high-level abstract facts. For that purpose, instead of relying on features extracted from the images, we used facts in the form of encoded objects detected by the Faster R-CNN detector. Those abstract facts were next passed to a reasoning module, developed with the aim of learning object-object relations. The achieved overall accuracy is comparable with the current state-of-the-art results. Analysis of the results unveiled that the proposed solution gives more stable results for different tasks, and, moreover, shows improvement in 2 out of 5 CLEVR tasks. In particular, it gives significant, few percent improvement in the Counting task, which is currently considered as one of the most complex tasks in VQA and is gaining more and more attention, e.g. in [39] the authors proposed a new dataset called HowMany-QA, devoted only to that specific problem.

The detailed analysis of the operation of components of our system has proven that OD is working properly, whereas difference in accuracy comes from the Relational Network module. According to the results reported in [35] the RN module working in separation is supposed to give around 2% better overall accuracy than the accuracy that we have managed to achieve. We treat that as a general promise that RN operating on more abstract features should be able to achieve even better results.

As the used reasoning module is the bottle-neck of the system, in our future works we want to focus on this part of the system and experiment with other approaches. The most important research directions, aiming at improvement of our system and overcoming its limitations, are as follows.

The first limitation is the used aggregation of relations detected by MLPs – as the number of objects in scenes might vary, utilization of recurrent neural net seems to be natural. An interesting solution to that problem was recently proposed in [31], where the authors developed a novel neural-based message passing mechanism called Recurrent Relational Networks and have shown its superiority on Sudoku puzzles and the BaBi textual QA. Second limitation of our system is that the reasoning performed in the Semantic Embedding module is strictly sequential, with hardcoded number of reasoning steps, whereas number of reasoning steps should in fact vary, depending on the task given. One of the possible solutions addressing that problem is to employ the Adaptive Computation Time (ACT) mechanism proposed in [12]. Yet another promising direction is utilization of external memory (i.e. memory augmented neural networks such as Neural Turing Machine [13]) for learning more complex reasoning schemes and memorization of the abstract, graph like representation of the observed scene before generating the final answer. The latter is a natural extension for reasoning over explicit high-level representations of the contents of the image and shows very good results as reported e.g. in [41]. Finally, we also want to improve our model by training it jointly on several tasks (e.g. word-level modeling, object detection and VQA) – it was recently shown e.g. in [20] that such a multi-task learning enables the model to develop unified representation and results in improved overall accuracy.

Acknowledgements

We would like to thank Alexis Asseman for setting up our hardware and managing our servers, so that we can run our experiments easily, Ahmet S. Ozcan for his insights and proofreading the paper, T.S. Jayram, Vincent Albouy, Ben Lyo and other members of our Machine Intelligence team in IBM Research, Almaden, for critical feedback and discussions.

References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. *arXiv preprint arXiv:1707.07998*, 2017.
- [3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Deep compositional question answering with neural module networks. *arXiv preprint arXiv:1511.02799*, 2, 2015.
- [4] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual Question Answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.
- [5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [6] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [7] B. Dai, Y. Zhang, and D. Lin. Detecting visual relationships with deep relational networks. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2017.
- [8] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human attention in visual question answering: Do humans and deep networks look at the same regions? *arXiv preprint arXiv:1606.03556*, 2016.
- [9] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.
- [10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2016.
- [11] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *arXiv preprint arXiv:1612.00837*, 2016.
- [12] A. Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.
- [13] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [15] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182. International World Wide Web Conferences Steering Committee, 2017.
- [16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [17] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. Speed/accuracy trade-offs for modern convolutional object detectors. *arXiv preprint arXiv:1611.10012*, 2016.
- [18] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997. IEEE, 2017.
- [19] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Inferring and executing programs for visual reasoning. *arXiv preprint arXiv:1705.03633*, 2017.
- [20] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.
- [21] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.
- [22] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowd-sourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.
- [23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [24] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.
- [26] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.
- [27] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in Neural Information Processing Systems*, pages 1682–1690, 2014.
- [28] M. Malinowski and M. Fritz. Towards a visual turing challenge. *arXiv preprint arXiv:1410.8027*, NIPS Workshop on Learning Semantics, 2014.
- [29] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A deep learning approach to visual question answering. *International Journal of Computer Vision*, 125(1-3):110–135, 2017.
- [30] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- [31] R. B. Palm, U. Paquet, and O. Winther. Recurrent relational networks for complex relational reasoning. *arXiv preprint arXiv:1711.08028*, 2017.

- [32] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [33] M. Ren, R. Kiros, and R. Zemel. Image question answering: A visual semantic embedding model and a new dataset. *Proc. Advances in Neural Inf. Process. Syst.*, 1(2):5, 2015.
- [34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- [35] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. *arXiv preprint arXiv:1706.01427*, 2017.
- [36] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4613–4621, 2016.
- [37] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgb-d images. *Computer Vision—ECCV 2012*, pages 746–760, 2012.
- [38] D. Teney, Q. Wu, and A. van den Hengel. Visual question answering: A tutorial. *IEEE Signal Processing Magazine*, 34(6):63–75, 2017.
- [39] A. Trott, C. Xiong, and R. Socher. Interpretable counting in visual question answering. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.
- [40] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel. FVQA: Fact-based visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.
- [41] P. Wang, Q. Wu, C. Shen, and A. v. d. Hengel. The vqa-machine: Learning how to use existing vision algorithms to answer new questions. *arXiv preprint arXiv:1612.05386*, 2016.
- [42] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.
- [43] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 2017.
- [44] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4622–4630, 2016.
- [45] F. Xiao, L. Sigal, and Y. J. Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2017.
- [46] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.
- [47] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016.
- [48] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank description generation and question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2461–2469, 2015.