Re-implementing and Extending Relation Network for R-CBIR

Nicola Messina(✉), Giuseppe Amato, and Fabrizio Falchi

ISTI-CNR, Pisa, Italy
{nicola.messina,giuseppe.amato,fabrizio.falchi}@isti.cnr.it

Abstract. Relational reasoning is an emerging theme in Machine Learning in general and in Computer Vision in particular. Deep Mind has recently proposed a module called Relation Network (RN) that has shown impressive results on visual question answering tasks. Unfortunately, the implementation of the proposed approach was not public. To reproduce their experiments and extend their approach in the context of Information Retrieval, we had to re-implement everything, testing many parameters and conducting many experiments. Our implementation is now public on GitHub and it is already used by a large community of researchers. Furthermore, we recently presented a variant of the relation network module that we called Aggregated Visual Features RN (AVF-RN). This network can produce and aggregate at inference time compact visual relationship-aware features for the Relational-CBIR (R-CBIR) task. R-CBIR consists in retrieving images with given relationships among objects. In this paper, we discuss the details of our Relation Network implementation and more experimental results than the original paper. Relational reasoning is a very promising topic for better understanding and retrieving inter-object relationships, especially in digital libraries.

Keywords: Relation Network · Image retrieval · Deep Learning · Visual features

1 Introduction

In the growing area of Computer Vision (CV), state-of-the-art Deep Learning methods show impressive results in tasks such as classifying or recognizing objects in images. Several recent studies, however, demonstrated the difficulties of such architectures to intrinsically understand a complex scene to catch spatial, temporal and abstract relationships among objects.

One of the most prominent fields of Deep Learning applied to CV within which these ideas are being tested is Relational Visual Question Answering (R-VQA). This task consists in answering to a question asked on a particular input image. While in standard VQA the question usually concerns single objects and their attributes, the R-VQA questions inquire about relationships between multiple objects in the image.

M. Ceci et al. (Eds.): IRCDL 2020, CCIS 1177, pp. 82–92, 2020.
https://doi.org/10.1007/978-3-030-39905-4_9

R-VQA is considered a challenging task for current state-of-the-art deep learning models since it requires a range of different reasoning capabilities. In fact, in addition to finding and classifying objects inside the image or understanding the meaning of each word of the input question, it is necessary to understand what are the relationships connecting visual objects and it is required to link together learned textual and visual representations.

This work is about implementing and training the Relation Network architecture (RN) [17]. Our final goal was to extend the RN to extract visual relationship-aware features for a novel task that we called Relational Content-Based Image Retrieval (R-CBIR). The R-CBIR task consists in finding all the images in a dataset that contains objects in similar relationships with respect to the ones present in a given query image.

Specifically, in [13] and [14] we introduced some extensions to the original RN module, able to extract visual relationship-aware features for efficiently characterizing complex inter-object relationships. We trained our RN variants on the CLEVR R-VQA task and we demonstrated that the extracted visual features were suitable for the novel R-CBIR task.

The high-level relational understanding could become a fundamental building block in digital libraries, where multi-modal information has to be processed in smart and scalable ways. Furthermore, R-CBIR encourages the development of solutions able to produce efficient yet powerful relationships-aware features, capable of efficiently describing the large number of inter-object relationships present in a digital library. A digital library, in fact, is composed of a large amount of multi-modal objects: it contains both multimedia elements (images, audio, videos) and text. One interesting challenge in digital libraries is finding relationships either between cross-domain data (e.g., a newspaper article with the related video in the newscast) or between the individual objects that are contained in a single multimedia element (e.g., the spatial arrangement of furniture in a picture of a room). This is a must for constructing strong and high-level interconnections between inter- and intra-domain data, to efficiently collect and manage knowledge.

The first step was re-implementing the RN architecture and training it on the CLEVR dataset, using the same setup detailed in the original work [17]. This was a necessary step since the original code was not published. RN was originally proposed by Deep Mind, a company owned by Google and our code is the first public working implementation of RN[1] on the CLEVR dataset. Thus, it is already largely used.

We found different issues during the replication process. Hence, in this paper, we give many details about the problems we addressed during the implementation of the original version of RN. In the end, we were able to successfully train this architecture reaching an accuracy of 93,6% on the CLEVR R-VQA task.

[1] https://github.com/mesnico/RelationNetworks-CLEVR.

2 Related Work

R-VQA. R-VQA comes from the task of VQA (Visual Question Answering). Plain VQA consists in giving the correct answer to a question asked on a given picture, so it requires connecting together different entities coming from heterogeneous representations (text and visuals).

Some works [20,22] proposed approaches to standard VQA problems on datasets such as VQA [1], DAQUAR [11], COCO-QA [16].

Recently, there is the tendency to conceptually separate VQA and R-VQA. In R-VQA, in fact, images contain difficult inter-object relationships, and question are formulated in a way that it is impossible for deep architectures to answer correctly without having understood high-level interactions between the objects in the same image. Some datasets, such as CLEVR [5], RVQA [10], FigureQA [8], move the attention towards this new challenging task.

In this work, we address the R-VQA task by employing the CLEVR dataset. CLEVR is a synthetic dataset composed of 3D rendered scenes. It contains simple yet photorealistic 3D shapes, and it is suitable for testing out, in a fully controlled environment, the intrinsic relational abilities of deep neural networks.

On the CLEVR dataset, [17] and [15] proposed a novel architecture specialized to think in a relational way. They introduced a particular layer called Relation Network (RN), which is specialized in comparing pairs of objects. Objects representations are learned by means of a four-layer CNN, and the question embedding is generated through an LSTM. The overall architecture, composed of CNN, LSTM, and the RN, can be trained fully end-to-end, and it is able to reach superhuman performances. Other solutions [4,6] introduce compositional approaches able to explicitly model the reasoning process by dynamically building a reasoning graph that states which operations must be carried out and in which order to obtain the right answer.

To close the performance gap between interpretable architectures and high performing solutions, [12] proposed a set of visual-reasoning primitives that are able to perform complex reasoning tasks in an explicitly interpretable manner.

R-CBIR. On the R-CBIR task there was some experimentation using both CLEVR and real-world datasets. [7] introduced a CRF model able to ground relationships given in the form of a scene graph to test images for image retrieval purposes. However, this model is not able to produce a compact feature. They employed a simple dataset composed of 5000 images and annotated with objects and their relationships.

More recently, using the Visual Genome dataset, [21] implemented a large scale image retrieval system able to map textual triplets into visual ones (object-subject-relation inferred from the image) projecting them into a common space learned through a modified version of triplet-loss.

The works by [2,13,14] exploit the graph data associated with every image in order to produce ranking goodness metrics, such as nDCG and Spearman-Rho ranking correlation indexes. Their objective was evaluating the quality of the ranking produced for a given query, keeping into consideration the relational

content of every scene. In particular, our previous works [13,14] analyzed two architectures for extracting relational data by exploiting knowledge acquired through R-VQA.

2.1 Original Setup

The overall architecture and the initial hyper-parameters we used in our code come from the original paper. Following, we briefly review this original setup.

The Relation Network (RN) [17] approached the task of R-VQA and obtained remarkable results on the CLEVR dataset. RN modules combine input objects forming all possible pairs and applies a common transformation to them, producing activations aimed to store information about possible relationships among input objects. For the specific task of R-VQA, authors used a four-layer CNN to learn visual object representations, that are then fed to the RN module and combined with the textual embedding of the question produced by an LSTM, conditioning the relationship information on the textual modality. The core of the RN module is given by the following:

$$r = \sum_{i,j} g_\theta(o_i, o_j, q)\,, \tag{1}$$

where g_θ is a parametric function whose parameters θ can be learned during the training phase. Specifically, it is a multi-layer perceptron (MLP) network. o_i and o_j are the objects forming the pair under consideration, and q is the question embedding vector obtained from the LSTM module. The answer is then predicted by a downstream network f_ϕ followed by a softmax layer that outputs probabilities for every answer:

$$a = softmax(f_\phi(r))\,. \tag{2}$$

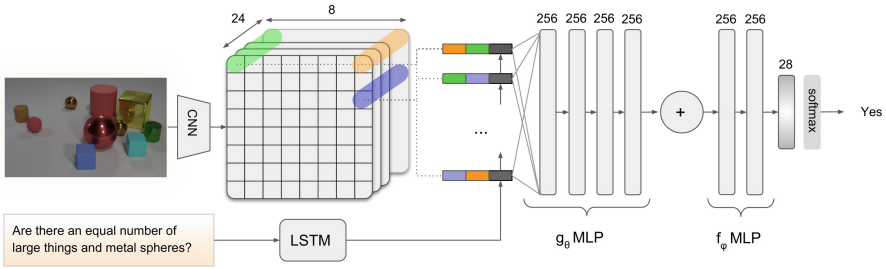

Fig. 1. Relation Network (RN) architecture.

During our implementation, we followed the guidelines and the hyper-parameters configuration by the authors. In particular, we setup the architecture as follows (Fig. 1 depicts the overall architecture):

- the CNN is composed of 4 convolutional layers each with 24 kernels, ReLU non-linearities and batch normalization;
- g_θ and f_ϕ are multilayer perceptrons. They are composed of 4 and 2 fully-connected layers of 256 neurons each. Every layer is followed by the ReLU non-linearity;
- a final linear layer with 28 units produces logits for a softmax layer over the answers vocabulary;
- dropout with 50% dropping probability is inserted after the penultimate layer of f_ϕ;
- the training is performed using the Adam optimizer, with a learning rate of $1e^{-4}$.

We took some decisions that probably brought our code to differ substantially from the original authors implementation. Concerning question processing, we built the dictionaries by sequentially scanning all the questions in the dataset. The zero index was used as padding during the embedding phase. We assumed uni-directional LSTM for question processing. Also, in the first place, we did not consider learning rate schedules nor dataset balancing procedures.

3 Preliminary Results

When training using the original configuration, we reached an accuracy plateau at around 53% on the validation set, while the authors claimed an accuracy of 95,5%.

We broke down the accuracy for the different question types, to have a better insight of what the network was learning. The validation accuracy curves are reported in Fig. 2.

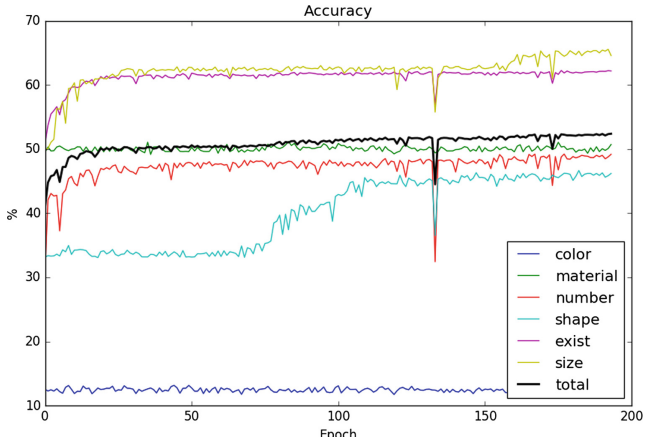

Fig. 2. Validation accuracy curve during the initial training.

This plot shows that the trained model was completely blind to the concepts of color and material since their accuracy was perfectly compatible with uniform random outcomes. However, even considering the other question types, the model was not performing as expected.

These results motivated us to concentrate on some implementation-level details that could help the network convergence.

In particular, we collected a list of some critical implementation details that may have played a role in the network training failure:

- **punctuation-level tokenization:** initially, we did not consider the punctuation as separate elements in the dictionary, so that sentences like *"There is a small cube behind the red object. What is its color?"* generated words like *"object."* and *"color?"*. Instead, one possibility was to break them down into four different entries: *"object"*, *"."*, *"color"*, and *"?"*.
- **training regularization:** in order to regularize training, we thought of adopting standard regularization procedures, like weight-decay and gradient clipping.
- **question inversion:** even if this trick applies to sentence translation models [19], it has been observed that feeding the LSTM with the reversed sentence often brings to an overall higher accuracy.
- **SGD optimizer:** the SGD optimizer is overall slower but asymptotically often performs slightly better than Adam.
- **answers balancing:** the answers distribution is not uniform in the CLEVR dataset. This could have caused problems during the training since less likely examples were penalized. One possible solution was trying to build batches in which all the answers were equally likely.
- **CNN pretraining:** to help the whole architecture to converge faster, we though of initializing the CNN parameters independently, by employing an easier non-relational task. In particular, we trained the CNN using a multi-label classification task, whose aim was to find out the attributes of all the objects inside the CLEVR scene. We aimed to bring the CNN parameters in a zone of the parameters space suitable for the downstream R-VQA task.
- **learning rate and batch size schedulers:** according to some detailed research on neural network parameters optimization, schedulers have a key role during training. We managed to try different schedulers to see if they could move the network parameters away from local minima.

4 Improvements

Following, we report our findings after experimenting with different variations of the original implementation.

Punctuation-Level Tokenization. First of all, we implemented the punctuation-level tokenization for processing the input questions. However, we immediately measured a strong drop in the validation accuracy, from 53% to 20%. This could be due to the fact that the network was effectively using the word-punctuation

tokens (e.g. *"color?"*) for easily discerning the question type and better attending to the key question details.

Training Regularization. Gradient clipping and weight decay helped to stabilize the training. However, they did not change the accuracy in any significant way.

Question Inversion. We tried feeding the questions to the LSTM in reverse order. With these changes, accuracy moved from 53% to around 66%. It turned out that the question inversion was a key implementation detail.

To understand how the network outputs were distributed after these changes, we prepared the confusion matrix measured on the validation accuracy (Fig. 3).

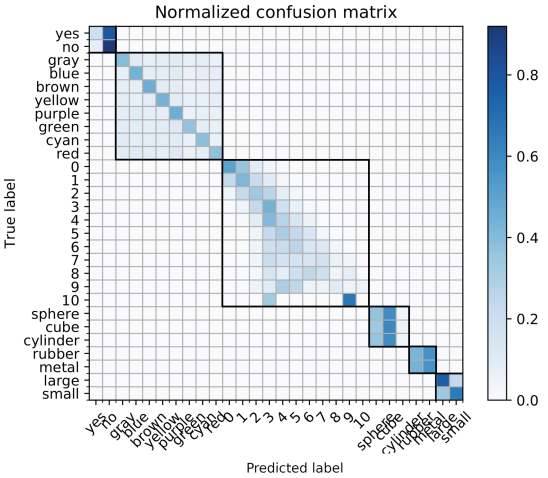

Fig. 3. Confusion matrix after question inversion.

In CLEVR scenario, there are 28 possible answers and they are clustered in 6 classes: *numbers*, *size*, *color*, *material*, *shape*, *exists*. In the confusion plot, there are 6 diagonal blocks corresponding to these classes. Empty entries outside the squared diagonal blocks show that answers falling outside their class were extremely unlikely. This was an important finding: the network was perfectly able to understand what kind of answer should be given in output (e.g. a binary yes/no answer rather than a color), but it was not able to figure out the correct label within that class.

SGD Optimizer. We tried training the network using the SGD optimizer, using the same learning rate employed with Adam ($1e^{-4}$). Unfortunately, the training process using SGD was too slow to collect some useful insights. In particular, during the first 50 epochs the architecture remained completely unable to solve the *number* and the *color* classes. Also, the model trained with SGD was not able to understand the 6 different question types, while with Adam this happened already during the first 5–7 epochs.

CNN Pretraining. Although the CNN pretraining sped up the overall convergence during the first epochs, it did not improve the overall validation accuracy. This made us formulate the hypothesis that the problem could be not in the perception module, but rather in the reasoning one, probably in the core of the relation network. In fact, the multi-label classification task reached a mean average precision of 0.99, meaning that the CNN was perfectly able to attend to all the object attributes. The multi-label classification task was trained using 5000 and 750 training and validation images respectively.

Answers Balancing. We wrote a custom batch sampler to ensure a uniform distribution among the answers. In this scenario, we obtained a better accuracy distribution among the different answer classes, w.r.t. the initial validation curves in Fig. 2. In particular, we observed that the model was no more color blind. However, once converged, the overall mean accuracy remained the same as in the initial experiment.

Schedulers. Initially, we tried standard learning rate schedulers, such as CosineAnnealing [9], Exponential, Step-Exponential, and ReduceOnPlateau. Unfortunately, none of them resulted in an accuracy boost, even trying different hyper-parameters such as the step size (in epochs) and the step multiplier.

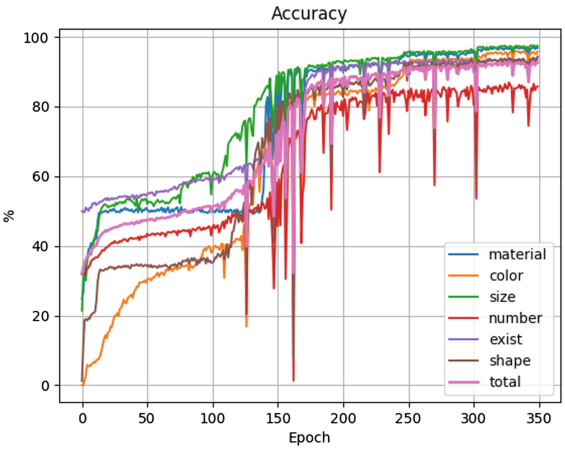

Fig. 4. Validation accuracy - increasing learning rate policy.

Accuracy started growing when we adopted the findings by [18], which suggested increasing the batch size instead of decreasing the learning rate. Our policy consisted in doubling the batch size every 45 training epochs, starting from 32 up to a maximum of 640. We experimented on the *state description* version of the dataset, in which the scene is already encoded in a tensor form suitable for

the relation network so that the perception pipeline (the CNN module) is temporarily kept apart. During this experiment, the learning rate remained fixed. With this batch size scheduling policy, we obtained an accuracy of 85%.

The best result, however, was reached using a warm-up learning rate scheduling policy similar to the one used in [3]. In particular, we doubled the learning rate every 20 epochs, from $1e^{-6}$ to $1e^{-4}$. When experimenting on the *state description* version of CLEVR, we were able to reach an accuracy of 97,9%. We repeated the same experiment on the full CLEVR, training end-to-end from pixels and words to answers, and we finally obtained an accuracy of 93,6%. This value is fully compatible with the accuracy claimed by the authors of 95,5%. Validation curves from this training setup are reported in Fig. 4.

The final confusion matrix in Fig. 5 highlights the answers for which there are still problems. Overall, the only remaining issues reside in the *number* class. In fact, the network has still some difficulties when the objective for a particular question is counting many object instances (the number *9* is almost never output as answer).

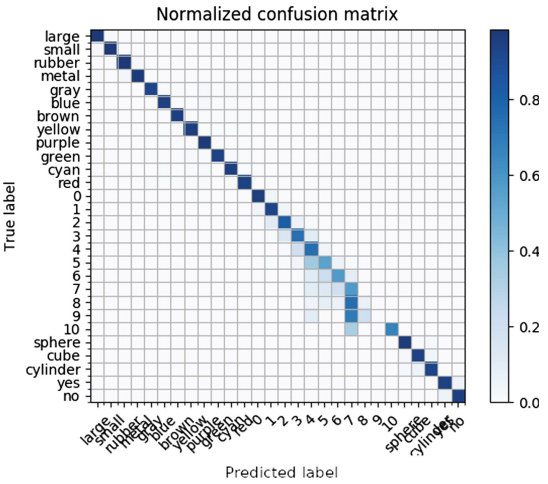

Fig. 5. Final confusion matrix.

5 Conclusions

In this work, we re-implemented the Relation Network architecture [17]. After a few experimentations, we were not able to reach the accuracy claimed by the authors for the R-VQA task on the CLEVR dataset. For this reason, we conducted multiple experiments testing different architectural and implementation tweaks to make the network converge to the claimed accuracy values. In the end, we discovered that the learning rate warm-up scheduling policy was the

main missing component. We were able to reach an accuracy of 93,6%, perfectly compatible with the one reached by the Deep Mind team.

We used these results to develop some extensions of the original Relation Network, capable of producing relationships-aware visual features for the novel task of R-CBIR. We noticed that slight modifications to the original architecture to achieve our R-CBIR objectives did not affect the network convergence when using the described learning rate scheduling policy. In particular, in [13] the two-stage RN (2S-RN) reached almost the same accuracy as the original architecture.

Instead, the introduction of the in-network visual aggregation layer in the Aggregated Visual Features RN (AVF-RN) architecture [14] made the performance drop to around 65%. This was due to the strong visual features compression needed. However, we demonstrated that AVF-RN was still able to produce state-of-the-art relationships-aware visual features suitable for R-CBIR.

References

1. Antol, S., et al.: VQA: visual question answering. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2425–2433 (2015)
2. Belilovsky, E., Blaschko, M.B., Kiros, J.R., Urtasun, R., Zemel, R.: Joint embeddings of scene graphs and images. ICLR (2017)
3. Goyal, P., et al.: Accurate, large minibatch SGD: training imageNet in 1 hour. http://arxiv.org/abs/1706.02677 (2017)
4. Hu, R., Andreas, J., Rohrbach, M., Darrell, T., Saenko, K.: Learning to reason: end-to-end module networks for visual question answering. In: The IEEE International Conference on Computer Vision (ICCV) (October 2017)
5. Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C.L., Girshick, R.: CLEVR: a diagnostic dataset for compositional language and elementary visual reasoning (2017)
6. Johnson, J., et al.: Inferring and executing programs for visual reasoning. In: The IEEE International Conference on Computer Vision (ICCV) (October 2017)
7. Johnson, J., et al.: Image retrieval using scene graphs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3668–3678 (2015)
8. Kahou, S.E., Atkinson, A., Michalski, V., Kádár, Á., Trischler, A., Bengio, Y.: FigureQA: an annotated figure dataset for visual reasoning. CoRR abs/1710.07300 (2017). http://arxiv.org/abs/1710.07300
9. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: ICLR (2017)
10. Lu, P., Ji, L., Zhang, W., Duan, N., Zhou, M., Wang, J.: R-VQA: learning visual relation facts with semantic attention for visual question answering. In: SIGKDD 2018 (2018)
11. Malinowski, M., Fritz, M.: A multi-world approach to question answering about real-world scenes based on uncertain input. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K. (eds.) Advances in Neural Information Processing Systems 27, pp. 1682–1690. Curran Associates, Inc. (2014)
12. Mascharka, D., Tran, P., Soklaski, R., Majumdar, A.: Transparency by design: closing the gap between performance and interpretability in visual reasoning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)

13. Messina, N., Amato, G., Carrara, F., Falchi, F., Gennaro, C.: Learning relationship-aware visual features. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11132, pp. 486–501. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11018-5_40
14. Messina, N., Amato, G., Carrara, F., Falchi, F., Gennaro, C.: Learning visual features for relational CBIR. Int. J. Multimedia Inf. Retr. 1–12 (2019). https://doi.org/10.1007/s13735-019-00178-7
15. Raposo, D., Santoro, A., Barrett, D.G.T., Pascanu, R., Lillicrap, T., Battaglia, P.W.: Discovering objects and their relations from entangled scene representations. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Workshop Track Proceedings (2017). https://openreview.net/forum?id=rkrjrvmKl
16. Ren, M., Kiros, R., Zemel, R.: Exploring models and data for image question answering. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2953–2961. Curran Associates, Inc. (2015)
17. Santoro, A., et al.: A simple neural network module for relational reasoning. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30, pp. 4967–4976. Curran Associates, Inc. (2017)
18. Smith, S., Kindermans, P.J., Ying, C., Le, Q.V.: Don't decay the learning rate, increase the batch size (2018). https://openreview.net/pdf?id=B1Yy1BxCZ
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27, pp. 3104–3112. Curran Associates, Inc. (2014). http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf
20. Yang, Z., He, X., Gao, J., Deng, L., Smola, A.: Stacked attention networks for image question answering. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2016)
21. Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A., Elhoseiny, M.: Large-scale visual relationship understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 9185–9194 (2019)
22. Zhou, B., Tian, Y., Sukhbaatar, S., Szlam, A., Fergus, R.: Simple baseline for visual question answering. CoRR abs/1512.02167 (2015). http://arxiv.org/abs/1512.02167