Visual Question Answering on Balanced Binary Datasets

Lawin Khalid
Project report for Artificial Intelligence: Cognitive Systems
University of Gothenburg
guskhala@student.gu.se

Abstract

This paper presents visual question answering experiments on binary datasets with different characteristics with regard to degree of balance and image type. We address the importance of a truly balanced dataset by making use of a semi-balanced dataset with clipart scenes, an unbalanced subset of real images, and a truly balanced subset of real images. Since a pretrained model, trained on real images, is used for representation of image features, this selection of datasets also examines how well these features applies to clipart images. Moreover, we examine the datasets' sensitivity to excluding visual features, and their sensitivity to altering the language representation by using GloVe vectors in some of the models. Findings highlight the importance of a truly balanced dataset, and shows that features from a pretrained model can also be exploited for use on clipart images. We also show that training on truly balanced data can improve the performance, likely because it forces the model to exploit visual features more efficiently.

1 Introduction

Visual question answering (VQA) is a current and challenging task in the field of Artificial Intelligence. This multi-discipline task combines computer vision, natural language processing, knowledge representation and reasoning. Earlier efforts of VQA have used small and unbalanced datasets, that create undesirable biases in the language part of the data. Malinowski and Fritz (2014), and Malinowski et al. (2015) use the DAQUAR dataset with 1,449 real world images and 12,468 question-answer pairs. Antol et al. (2015) colleced the very large VQA dataset, including approximately 250,000 real word and clipart images and 765,000 question-answer pairs of various types, such as binary (yes/no) questions, number questions, color questions, among others. This dataset, however, remains unbalanced, which results in language biases in the data. Some examples can be extreme. Answering *yes* on all questions starting with *Is there a clock*, would give 98% accuracy (Goyal et al., 2017). Lately, efforts have been made to balance the clipart images of the VQA dataset (Zhang et al., 2016), and later the real images (Goyal et al., 2017). See section 2 for more detailed information on this balancing process. The present study is based on these two balanced datasets.

In this project, we focus on binary questions for several reasons. One reason is simplicity. Binary questions offer easy evaluation and a clearly defined baseline, as well as possibility to make wider experiments, due to the simple nature of the questions and answers, while still fitting within the scope of this project. Secondly, we are interested in exploring whether clipart images and real images behave differently when used with the pretrained ResNet50 CNN, that has been trained on real images. As of today, this is only possible with binary questions, since the balanced abstract scenes database does not contain any other question types. Lastly, and more importantly, binary questions are the only type of question at this point that we can make *truly balanced*, in a simple enough way to fit the scope of this project. That is, for every given question, *all* answers in the data should have the same frequency, to overcome any language bias in the data. The datasets were balanced by collecting an image and *one* answer that is different from the original answer, for each question. Hence, the only question type that becomes truly balanced with this method is a ques-

tion type with only two possible answers, namely binary questions.

The goal is to make experiments with three different models on these three datasets with various characteristics. The model architecture is CNN + LSTM. We use one simple model without GloVe word embeddings and one model with GloVe word embeddings. This allows us to draw conclusion about the value of pretrained GloVe word embeddings in VQA. In addition, we use one language-only model, which allows us to observe how sensitive the variously balanced datasets are to removal of visual features, and consequently draw conclusions about to what degree the models exploit language biases when present. Both models that use visual features take as image representation the penultimate layer of ResNet50, a pretrained 50 layer deep CNN model, trained on real images. We are interested in exploring whether this representation can be successfully exploited also on the clipart images. In total nine experiments are made.

2 Datasets

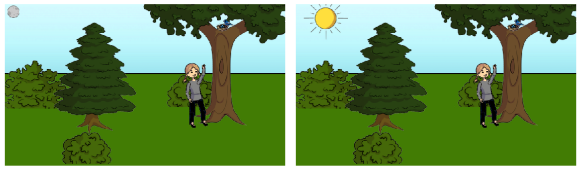
The two datasets used in the experiments, Balanced Real Images, and Balanced Binary Abstract Scenes, are both available at the visualqa homepage.[1] Both these datasets were created as a further development of the original VQA project (Antol et al., 2015). The real world images were collected from the MSCOCO database (Lin et al., 2014), and annotated by Amazon Mechanical Turk (AMT) workers. The abstract scenes have the benefit of allowing to focus on high-level semantic reasoning in contrast to low-level reasoning, and is therefore well suited for VQA tasks, according to Antol et al. (2015) and Zhang et al. (2016). They were created and annotated by AMT workers, with directions from the authors (Antol et al., 2015). Both datasets were made balanced with the help of AMT workers, but there are some differences in the procedure.

The abstract scenes dataset were balanced by giving the AMT workers the task of altering the original image as little as possible, while at the same time changing the answer to the question from *yes* to *no*, or from *no* to *yes*. This was possible due to the nature of the clipart images, where workers can freely alter the scenes. However, this also resulted in that some scenes were not possible to create complementary scenes for. One example could be the question *Is it raining?*, accompanied by an indoor scene. Due to this constraint, the dataset is not truly balanced, but nevertheless more balanced than the original version of the abstract scenes dataset. These unbalanced scenes represent 5.93% of the total dataset, and for unknown reason, the authors still decided to include these scenes in the new dataset. In addition, in 14.55% of the examples, the human annotated answers did not match with the intended answer, which further imbalances the dataset. The total ratio of unbalanced examples are 20.48% of the dataset. (Zhang et al., 2016)

The real images dataset was instead balanced by finding completely new images with a different answers. This was done by first computing the 24 nearest neighbours of each image, by taking image representations of the images from the penultimate layer of a deep CNN and then finding the neighbours using l_2-distance. The 24 nearest neighbours were then presented to AMT workers, with the task of choosing an image for which the answer to the original question would not be the same as the original answer. Also this approach created some issues. For 22% of the questions in the dataset, the workers found no image among the 24 neighbours that fulfilled the criteria, thus choosing the provided option *not possible*. Moreover, similar to the abstract scenes dataset, in 9% of the examples, the human annotated answers did not match the intended answers. Also the authors of the real images dataset included the unbalanced examples in the new version of the dataset. Consequently, a total of 31% of the real images dataset was not balanced. (Goyal et al., 2017)

Examples of complementary pairs in the datasets are presented in Figure 1 and Figure 2. The real images dataset was first arbitrarily truncated at the length of the abstract scenes dataset, so that both datasets had the same size. The real images dataset was not arranged so that complementary pairs followed each other in order, but instead arranged in a way where consecutive questions referred to the same image. This led to the assumption that the subset was no longer balanced. The results later confirmed this. For this reason, a second subset was created from the real images, where only truly balanced examples were included. This procedure is further explained in section 3.1. For the sake of this project, the ab-

[1] https://visualqa.org/



Yes No

Figure 1: Complementary images for the question *Is it night time* in the abstract scenes dataset.

Yes No

Figure 2: Complementary images for the question *Is the chair in a dining room* in the real images dataset.

stract scenes dataset is considered to be *semi-balanced*, the arbitrarily truncated subset of the real images dataset is considered *unbalanced*, and the second subset of the real images is considered *truly balanced*. The datasets each contain 22,055 question-answer pairs, with accompanied images in the training set (22,056 for the truly balanced real images subset, due to the requirement of complementary pairs for every question), and 11,326 question-answer pairs with images in the validation set. While the real images dataset originally included a training set, a validation set, and a test set, the abstract scenes dataset did not include any test set. For this reason, the validation set was used for evaluation for both datasets, and the intended test set of the real images dataset was omitted.

3 Method

3.1 Preparing the datasets

We begin by preparing the datasets for the experiments. Some help was gotten by making use of the VQA API[2], which the file `prepare_data.ipynb` is based on. The API allowed us to select only binary questions from the real images dataset in a straight-forward manner, with the help file `vqa.py`. We first load the original metadata from the json-files, and filter out non-binary questions with the API. Since the abstract scenes dataset only includes binary questions, this step was only applied to the real images dataset.

Next, we collect the question, the most common answer and the image file name, for every example, and save these to separate, aligned files. This was done by adding a new function, `writeQA`, to the help file `vqa.py`. In this function, we also create new metadata in a similar manner to the VQA tutorial. That is, we create a vocabulary of the questions and answers, along with their count. Also, we measure the number of words in each sentence and save the max length for each dataset. The most common answer was taken directly from the json metadata, under the entry `multiple_choice_answer`. It is not clear how the original authors treated cases with identical frequency of *yes* and *no*.

It was found that there were still some answers that were not *yes* or *no*. This was true for both datasets. For example, in the vocabulary of the answers of the real images dataset we found the words *africa*, *not*, *cutting*, *apples*, and *cups*. The max length in the answers was 2. We can assume that a question like *Are the apples green?*, accompanied by an image of pears would provoke an answer like *not apples*. Since these examples were very few, it was judged safe to filter them out. Also this was done in the `writeQA` function. The aforementioned steps were repeated for the training and validation sets of both datasets.

The real images dataset was, in contrast to the abstract scenes dataset, not sorted by complementary pairs, but instead sorted in a way where questions accompanied by the same image followed each other. To make a truly balanced real images dataset with the same size of the abstract scenes dataset, we therefore have to sort the data with the help of a complimentary pairs list, available at the VQA homepage. This list provides the annotation ids of the complementary pairs. With the use of the VQA API, this could be used to map the ids to questions, answers and images. The procedure was to read the complementary pairs list, and collect the questions, answers and image file names of only the examples that also had a complementary example. These results were saved

[2] https://github.com/GT-Vision-Lab/VQA/tree/master/PythonHelperTools/

in a sorted manner, so that a future truncation would preserve the balance. For this, yet another function, `writeQA2`, was created in the help file `vqa.py`. Like its predecessor, this function also created new metadata including vocabulary, word frequency and max length of questions.

3.2 Models

The three models used, are explained further in this section. The models are based on the VQA tutorial by Mehdi Ghanimifard[3]. Since the aim was not to achieve as high score as possible, but rather to compare the results of different models on different datasets, no exhaustive effort was put to alter the model architecture. Moreover, no advanced techniques, such as visual attention, ware used for the visual features.

During training, 10% of the training data was used for validation. Generalization techniques included dropout and early stopping with patience of 10 epochs, to address overfitting. Moreover, we saved the best model with validation loss as monitor, and reduced the learning rate on plateau with a ratio of 0.2 and patience of 5 epochs. That is, whenever the validation loss did not decrease during 5 epochs, the learning rate was lowered by 80%. Training then continued for at least 5 more epochs before stopping and saving the model with lowest loss on the validation data. If at any time during these 10 epochs the validation loss did improve to a lower score, both patiences were reset. An exception to this technique was the simple model, where no reduce learning rate on plateau was used, and early stopping was used with a patience of 5 epochs.

The model architecture of the GloVe model is shown in figure 3. This is identical to the simple model, apart from its dimensions. The left and right branches before the concatenation represent the visual features and the language features respectively. The language-only model has only one input, and thus, the upper left branch, including the concatenation layer, is omitted in this model.

Simple model

Experiments with the simple model are done with `model_simple.ipynb`. We start by loading the data created with `prepare_data.ipynb`, as well as the pretrained ResNet50 CNN model

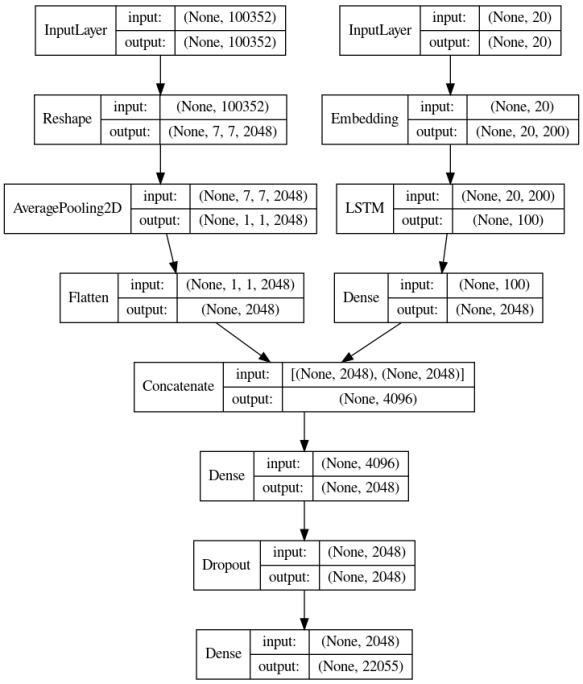


Figure 3: Model architecture.

for visual features. We also add three more entries to the vocabulary: `<pad>`, used for padding, `<unk>`, used for unknown and rare words, and `?`, used as an end-tag for the questions. However, the reasoning is that the `?` will not make any difference to the performance, but it was nevertheless kept from the original VQA tutorial. Moreover, we filter away words that only occur once in the vocabulary. This is done as a generalization technique to get better representations for unknown words in the validation set. We then preprocess the questions such that every word in the vocabulary gets represented by an integer, and represent each question as an array of integers, with the length of the longest sentence in the training set. For shorter sentences, we pad the beginning of the sentence (pre-padding). Trial runs showed no difference in performance between pre-padding and post-padding. Padding is required by Keras, so that all inputs have the same shape. For answers, we simply set the integer 1 to represent *yes*, and 0 to represent *no*. No padding is required, since we filtered away examples that did not have yes/no answers.

The steps are repeated for the validation set, but with use of the vocabulary from the training set. This sets `<unk>` to not only rare words, but also words that are not in the vocabulary of the training set. Moreover, the sentence length of the

[3] https://github.com/sdobnik/aics/tree/master/tutorials/vqa

training set is applied to the validation set. In the case where a validation question is longer than the longest sentence in the training set, we truncate the sentence from the end, risking that important words are omitted. Instead, it might be a good idea to extend the length of the arrays in both sets, to account for longer questions in the validation set. This was however not addressed in the present study.

Image representations were done by converting images to vectors, and then get feature representations from the penultimate layer of the pretrained ResNet50 model. This was done with the function from the original VQA tutorial. Moreover, the model architecture during training was kept as it was in the tutorial.

GloVe model

Experiments with the GloVe model are done with `model_glove.ipynb`. Image representations in the GloVe model were the same as in the simple model. What was changed here was instead the language representations. Firstly, we used Keras preprocessing tools for creating integer representations of words in the questions. It was found that this process was less sophisticated than the process in the original tutorial. Keras preprocessing tools did not treat rare and unknown words in a desirable manner. Instead of assigning a unique integer to these words, they were simply removed. When an unknown word occurred in the middle of the sentence, the created gap was closed by conjoining the surrounding words. Other than this, the preprocessing tool produced a similar representation of questions as in the simple model.

The main difference in this model is that pretrained weights are used in the embedding layer. The weights are obtained by using GloVe word embeddings (Pennington et al., 2014). We use the vectors trained on 6 billion words, vocabulary of 400,000 words, and 200 dimensions.[4] We create a dictionary, mapping each word in our vocabulary to its GloVe vector. We now have mapping from word to integer, created by Keras preprocessing tool, and mapping from word to GloVe vector. We combine these in a matrix, mapping each integer to the GloVe vector of the corresponding word. Each row of the matrix represents the word with that index as integer representation. This matrix is then used as weights for the embedding layer of the Keras model, which we freeze to protect it from being retrained.

Furthermore, we do a few alterations of the model. Firstly, we extend the LSTM output dimension from 50 to 100. We also try different dropout values of 0.2, 0.4, and 0.6 and use the value that produces the best model in terms of validation loss. The optimal value was 0.2 for the abstract scenes dataset and the truly balanced real images dataset, and 0.6 for the unbalanced real images dataset. This gives us an early indication that the unbalanced dataset is more prone to overfitting. We also use the callback reduce learning rate on plateau, as described in section 3.2.

Language only model

In the language-only model no visual features was used during training or testing. Otherwise, the model was identical to the GloVe model, with the same preprocessing procedure for the language features and the same settings for the hyper-parameters. This allowed us to give a reliable comparison with the full GloVe model. The experiments are done in the file `model_glove_no-vis-feat.ipynb`.

4 Results and discussion

Results are shown in Table 1. We observe all values, but mainly consider weighted average F1-score as the most robust metric for cross-model, cross-dataset comparison, as it takes most variables into account, including differences in answer distribution across the datasets. The VQA metric is the widely used metric for the VQA datasets, and is presented on the VQA homepage. It is calculated in a way that takes inter-human variability into account, shown in equation (1). This metric should not be used for cross-dataset comparison. A higher score could simply mean that there is more inter-human variation in that dataset. The provided VQA evaluation tool[5] was not compatible with the subsets used in the present study. For that reason, a custom code was created in the file `vqa_eval_metric.ipynb`.

$$\mathrm{Acc}(ans) = \min\left\{\frac{\#\text{humans that said } ans}{3}, 1\right\} \quad (1)$$

We can clearly see that the balanced datasets (the semi-balanced abstract scenes dataset and the

[4] https://nlp.stanford.edu/projects/glove/

[5] https://github.com/GT-Vision-Lab/VQA/tree/master/PythonEvaluationTools

	Abstract scenes			Real images			Truly balanced real		
Metric	SIM	+GLV	-IMG	SIM	+GLV	-IMG	SIM	+GLV	-IMG
VQA metric	65.22	67.10	64.92	64.17	64.01	64.18	67.85	67.38	63.18
Accuracy	52.60	54.17	51.07	55.30	55.06	52.27	56.13	55.69	50.00
Weighted avg F1	49.12	54.09	36.70	55.13	53.31	52.01	55.55	55.69	44.62
True 'yes'	39.01	28.75	01.40	31.49	37.90	30.53	22.38	28.99	09.41
False 'yes'	37.21	25.38	01.13	24.81	31.47	26.89	16.25	22.29	09.41
True 'no'	13.59	25.42	49.66	23.81	17.16	21.74	33.75	27.71	40.59
False 'no'	10.20	20.46	47.80	19.89	13.47	20.85	27.62	22.01	40.59

Table 1: Results for the three datasets. SIM = simple model, +GLV = model with GloVe word embeddings as weights for language representation, -IMG = GloVe model with no visual features. Values in percent.

truly balanced real images dataset) suffer considerably more by removal of visual features, while the unbalanced dataset is not as sensitive. This confirms the importance of balanced data for VQA. Observing the prediction distribution in the bottom of the table, we see that the language-only model is clearly more confused on the balanced and semi-balanced datasets. For the abstract scenes dataset the model chose the majority answer *no*, for almost all predictions, indicating that, although not truly balanced, there were not much language bias in the data that was consistent across training and validation sets. One somewhat surprising result was that the semi-balanced abstract scenes dataset suffered more from removing visual features than the truly balanced real images dataset, when observing F1-score and prediction distribution. However, the validation accuracy on the truly balanced dataset were constantly at 50% during training; see figure 4. No other model/dataset showed this behaviour. This indicates that there were no valuable information whatsoever in the language features alone, and that the language+image models are successfully exploiting the combination of language and visual features.

While the unbalanced real images dataset also suffers from removal of visual features, we can see that it is not as sensitive as the balanced datasets. F1-score with the language-only model is significantly higher on the unbalanced dataset, indicating that there are language biases in the dataset to exploit. It is also visible in the prediction distribution, showing that the model is not as confused in its predictions.

Moreover, we find that features from the pre-

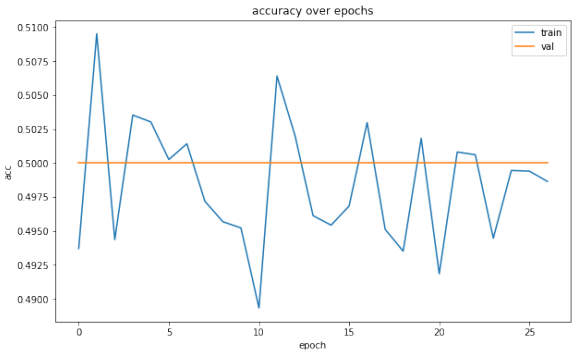


Figure 4: Accuracy during training with the language-only model on the truly balanced dataset.

trained ResNet50 CNN can be exploited for use also on clipart images. This finding is suggested by the fact that the abstract scenes dataset is very sensitive to removal of visual features, which indicates that the language+image models are indeed using the visual features from ResNet50 in their predictions.

The value of GloVe word embeddings as language representation are not consistent across datasets. While it strongly improves the performance on the abstract scenes dataset, it lowers the performance on the unbalanced real images dataset, and keeps the performance more or less unchanged on the truly balanced dataset. The reason for this behaviour is unclear at this stage. Due to time constraints during this project, no more effort was put to further examine this question.

A cross-dataset comparison of the results shows that the truly balanced have the best performance with the language+image models. This somewhat surprising finding suggests that the balance

forces the model to exploit visual features more efficiently. While the state-of-the-art models performed significantly worse on the semi-balanced datasets than on the unbalanced datasets (Zhang et al., 2016; Goyal et al., 2017), this cannot be confirmed on our truly balanced dataset. One hypothesis is that for the semi-balanced datasets, the model to some degree still rely on the language bias to build its weights, and that these biases are not consistent across training and validation set. When we remove all language biases, as in our truly balanced dataset, the model has to build its weights on more robust information. The reason that the models perform better on the unbalanced dataset, compared to the semi-balanced, can then be that there are more language biases in the unbalanced data, and consequently, the likelihood of finding the same biases in the validation data is larger. These are however just hypotheses, and should be examined further in future studies. We cannot confirm at this point whether the better performance on the unbalanced real images dataset compared to the semi-balanced abstract scenes dataset is due to different image types, or variation in balance, since both variables are changed between these datasets.

Furthermore, we see that the real images dataset outperforms the abstract scenes dataset in all models, in terms of accuracy, and in all models but the GloVe model in terms of weighted average F1-score. However, when observing the VQA metric, which takes inter-human variability into account, we see the opposite result. This indicates that the answers for the abstract scenes dataset have more variation among the 10 annotators, which suggests that annotators found the complementary pairs ambiguous.

5 Conclusions and future work

This study confirms the importance of overcoming language biases in VQA tasks. While recent efforts have been made to balance existing datasets, these cannot be considered truly balanced, as the authors opted to also include examples where the desired balancing failed. The models in our study clearly behaved differently on semi-balanced and truly balanced datasets. Semi-balanced datasets are a step in the right direction, but more efforts should be done to make the datasets truly balanced. That is, both excluding examples that were workers failed at finding or creating a complementary image, and examples where the human annotations were different from the expected answer. The trivial method used in the present study suffice for this purpose. Since our results indicate that only a truly balanced dataset forces the model to ground its answer in the visual content, more studies should be done to confirm this. There should also be focus on the quality of the complementary pairs. The method used to balance the abstract scenes database created ambiguous complementary pairs, where 14.55% of the examples did not get the human answers that the authors intended. Our results of the VQA metric also suggests that there are ambiguous complementary pairs in this dataset. Furthermore, there is reason to consider another metric for binary questions. Changing the denominator in equation (1) to a higher value would be more suitable for questions with few possible answers.

Other recent papers propose alternative ways of overcoming language priors in VQA tasks. Agrawal et al. (2018) propose new splits of the VQA datasets, changing the distribution of existing language priors between train and test sets for every question type (the question type of a question is determined by its first few words in the question), so that the same biases are not present across the sets. The concept idea is similar to the method used in the present study to create truly balanced data. Ramakrishnan et al. (2018) propose an interesting novel approach, introducing a game between a complete base VQA model and a language-only model which gets as input the question representation, encoded by the VQA model. The VQA model is then encouraged to alter its question embedding in order to reduce the performance of the language-only model, while maintaining its own performance. The result is that the model produces a representation that does not carry the language biases present in the data, and forces the model to ground its answers in the visual information instead.

We also find that a pretrained deep CNN model, such as ResNet50, can indeed be used as the visual part of a VQA model on clipart images, although trained on real world images. More research should be done to get a deeper understanding about the applicability of pretrained models on other image types. Regarding GloVe word embeddings, no stable conclusions could be drawn about its value in VQA tasks in our experiments.

References

Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. 2018. Don't just assume; look and answer: Overcoming priors for visual question answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4971–4980.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. *NIPS*.

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. 2015. Ask your neurons: A neural-based approach to answering questions about images. *CoRR*, abs/1505.01121.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. 2018. Overcoming language priors in visual question answering with adversarial regularization. *CoRR*, abs/1810.03649.

Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Yin and Yang: Balancing and answering binary visual questions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.