Ask Your Neurons: A Neural-based Approach to Answering Questions about Images

Mateusz Malinowski[1] Marcus Rohrbach[2] Mario Fritz[1]
[1]Max Planck Institute for Informatics, Saarbrücken, Germany
[2]UC Berkeley EECS and ICSI, Berkeley, CA, United States

Abstract

We address a question answering task on real-world images that is set up as a Visual Turing Test. By combining latest advances in image representation and natural language processing, we propose Neural-Image-QA, an end-to-end formulation to this problem for which all parts are trained jointly. In contrast to previous efforts, we are facing a multi-modal problem where the language output (answer) is conditioned on visual and natural language input (image and question). Our approach Neural-Image-QA doubles the performance of the previous best approach on this problem. We provide additional insights into the problem by analyzing how much information is contained only in the language part for which we provide a new human baseline. To study human consensus, which is related to the ambiguities inherent in this challenging task, we propose two novel metrics and collect additional answers which extends the original DAQUAR dataset to DAQUAR-Consensus.

1. Introduction

With the advances of natural language processing and image understanding, more complex and demanding tasks have become within reach. Our aim is to take advantage of the most recent developments to push the state-of-the-art for answering natural language questions on real-world images. This task unites inference of question intends and visual scene understanding with a word sequence prediction task.

Most recently, architectures based on the idea of layered, end-to-end trainable artificial neural networks have improved the state of the art across a wide range of diverse tasks. Most prominently Convolutional Neural Networks have raised the bar on image classification tasks [16] and Long Short Term Memory Networks are dominating performance on a range of sequence prediction tasks such as machine translation [28].

Very recently these two trends of employing neural architectures have been combined fruitfully with methods that can generate image [12] and video descriptions [30]. Both are conditioning on the visual features that stem from deep learning architectures and employ recurrent neural network approaches to produce descriptions.

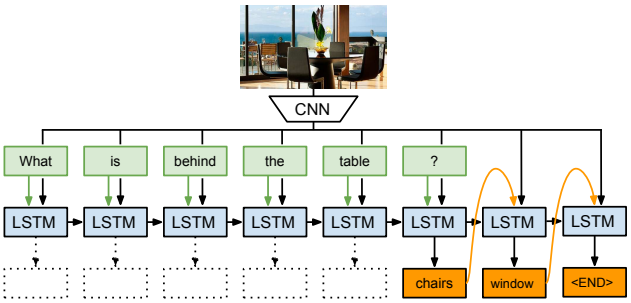

Figure 1. Our approach Neural-Image-QA to question answering with a Recurrent Neural Network using Long Short Term Memory (LSTM). To answer a question about an image, we feed in both, the image (CNN features) and the question (green boxes) into the LSTM. After the (variable length) question is encoded, we generate the answers (multiple words, orange boxes). During the answer generation phase the previously predicted answers are fed into the LSTM until the ⟨END⟩ symbol is predicted.

To further push the boundaries and explore the limits of deep learning architectures, we propose an architecture for answering questions about images. In contrast to prior work, this task needs conditioning on language as well visual input. Both modalities have to be interpreted and jointly represented as an answer depends on inferred meaning of the question and image content.

While there is a rich body of work on natural language understanding that has addressed textual question answering tasks based on semantic parsing, symbolic representation and deduction systems, which also has seen applications to question answering on images [20], there is initial evidence that deep architectures can indeed achieve a similar goal [33]. This motivates our work to seek end-to-end architectures that learn to answer questions in a single holistic and monolithic model.

We propose Neural-Image-QA, an approach to question

answering with a recurrent neural network. An overview is given in Figure 1. The image is analyzed via a Convolutional Neural Network (CNN) and the question together with the visual representation is fed into a Long Short Term Memory (LSTM) network. The system is trained to produce the correct answer to the question on the image. CNN and LSTM are trained jointly and end-to-end starting from words and pixels.

Contributions: We proposes a novel approach based on recurrent neural networks for the challenging task of answering of questions about images. It combines a CNN with a LSTM into an end-to-end architecture that predict answers conditioning on a question and an image. Our approach significantly outperforms prior work on this task – doubling the performance. We collect additional data to study human consensus on this task, propose two new metrics sensitive to these effects, and provide a new baseline, by asking humans to answer the questions without observing the image. We demonstrate a variant of our system that also answers question without accessing any visual information, which beats the human baseline.

2. Related Work

As our method touches upon different areas in machine learning, computer vision and natural language processing, we have organized related work in the following way:

Convolutional Neural Networks for visual recognition. We are building on the recent success of Convolutional Neural Networks (CNN) for visual recognition [16, 17, 25], that are directly learnt from the raw image data and pre-trained on large image corpora. Due to the rapid progress in this area within the last two years, a rich set of models [27, 29] is at our disposal.

Recurrent Neural Networks (RNN) for sequence modeling. Recurrent Neural Networks allow Neural Networks to handle sequences of flexible length. A particular variant called Long Short Term Memory (LSTM) [9] has shown recent success on natural language tasks such as machine translation [3, 28].

Combining RNNs and CNNs for description of visual content. The task of describing visual content like still images as well as videos has been successfully addressed with a combination of the previous two ideas [5, 12, 31, 32, 37]. This is achieved by using the RNN-type model that first gets to observe the visual content and is trained to afterwards predict a sequence of words that is a description of the visual content. Our work extends this idea to question answering, where we formulate a model trained to generate an answer based on visual as well as natural language input.

Grounding of natural language and visual concepts. Dealing with natural language input does involve the association of words with meaning. This is often referred to as grounding problem - in particular if the "meaning" is associated with a sensory input. While such problems have been historically addressed by symbolic semantic parsing techniques [15, 22], there is a recent trend of machine learning-based approaches [12, 13, 14] to find the associations. Our approach follows the idea that we do not enforce or evaluate any particular representation of "meaning" on the language or image modality. We treat this as latent and leave this to the joint training approach to establish an appropriate internal representation for the question answering task.

Textual question answering. Answering on purely textual questions has been studied in the NLP community [2, 18] and state of the art techniques typically employ semantic parsing to arrive at a logical form capturing the intended meaning and infer relevant answers. Only very recently, the success of the previously mentioned neural sequence models as RNNs has carried over to this task [10, 33]. More specifically [10] uses dependency-tree Recursive NN instead of LSTM, and reduce the question-answering problem to a classification task. Moreover, according to [10] their method cannot be easily applied to vision. [33] propose different kind of network - memory networks - and it is unclear how to apply [33] to take advantage of the visual content. However, neither [10] nor [33] show an end-to-end, monolithic approaches that produce multiple words answers for question on images.

Visual Turing Test. Most recently several approaches have been proposed to approach Visual Turing Test [21], i.e. answering question about visual content. For instance [8] have proposed a binary (yes/no) version of Visual Turing Test on synthetic data. In [20], we present a question answering system based on a semantic parser on a more varied set of human question-answer pairs. In contrast, in this work, our method is based on a neural architecture, which is trained end-to-end and therefore liberates the approach from any ontological commitment that would otherwise be introduced by a semantic parser.

We like to note that shortly after this work, several neural-based models [24, 19, 7] have also been suggested. Also several new datasets for Visual Turing Tests have just been proposed [1, 35] that are worth further investigations.

3. Approach

Answering questions on images is the problem of predicting an answer $\boldsymbol{a}$ given an image $\boldsymbol{x}$ and a question $\boldsymbol{q}$ according to a parametric probability measure:

$$\hat{\boldsymbol{a}} = \underset{\boldsymbol{a} \in \mathcal{A}}{\arg\max}\, p(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{q}; \boldsymbol{\theta}) \tag{1}$$

where $\boldsymbol{\theta}$ represent a vector of all parameters to learn and $\mathcal{A}$ is a set of all answers. Later we describe how we represent $\boldsymbol{x}$, $\boldsymbol{a}$, $\boldsymbol{q}$, and $p(\cdot|\boldsymbol{x}, \boldsymbol{q}; \boldsymbol{\theta})$ in more details.

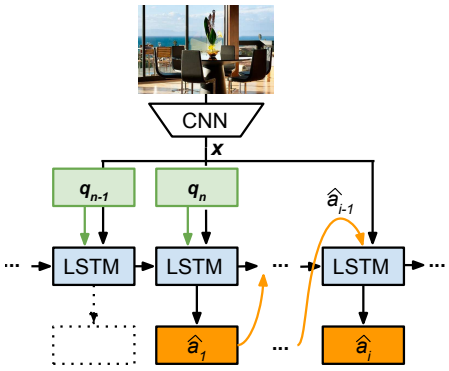

Figure 2. Our approach Neural-Image-QA, see Section 3 for details.

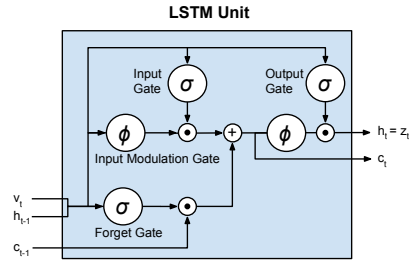

Figure 3. LSTM unit. See Section 3, Equations (3)-(8) for details.

In our scenario questions can have multiple word answers and we consequently decompose the problem to predicting a set of answer words $\boldsymbol{a}_{\boldsymbol{q},\boldsymbol{x}} = \{\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_{\mathcal{N}(q,x)}\}$, where $\boldsymbol{a}_t$ are words from a finite vocabulary $\mathcal{V}'$, and $\mathcal{N}(q,x)$ is the number of answer words for the given question and image. In our approach, named Neural-Image-QA, we propose to tackle the problem as follows. To predict multiple words we formulate the problem as predicting a sequence of words from the vocabulary $\mathcal{V} := \mathcal{V}' \cup \{\$\}$ where the extra token \$ indicates the end of the answer sequence, and points out that the question has been fully answered. We thus formulate the prediction procedure recursively:

$$\hat{\boldsymbol{a}}_t = \arg\max_{\boldsymbol{a}\in\mathcal{V}} p(\boldsymbol{a}|\boldsymbol{x}, \boldsymbol{q}, \hat{A}_{t-1}; \boldsymbol{\theta}) \tag{2}$$

where $\hat{A}_{t-1} = \{\hat{\boldsymbol{a}}_1, \ldots, \hat{\boldsymbol{a}}_{\boldsymbol{t-1}}\}$ is the set of previous words, with $\hat{A}_0 = \{\}$ at the beginning, when our approach has not given any answer so far. The approach is terminated when $\hat{a}_t = \$$. We evaluate the method solely based on the predicted answer words ignoring the extra token \$. To ensure uniqueness of the predicted answer words, as we want to predict the *set* of answer words, the prediction procedure can be be trivially changed by maximizing over $\mathcal{V} \setminus \hat{A}_{t-1}$. However, in practice, our algorithm learns to not predict any previously predicted words.

As shown in Figure 1 and Figure 2, we feed Neural-Image-QA with a question as a sequence of words, i.e. $\boldsymbol{q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_{n-1}, [\![?]\!]]$, where each $\boldsymbol{q}_t$ is the t-th word question and $[\![?]\!] := \boldsymbol{q}_n$ encodes the question mark - the end of the question. Since our problem is formulated as a variable-length input/output sequence, we model the parametric distribution $p(\cdot|\boldsymbol{x}, \boldsymbol{q}; \boldsymbol{\theta})$ of Neural-Image-QA with a recurrent neural network and a softmax prediction layer. More precisely, Neural-Image-QA is a deep network built of CNN [17] and Long-Short Term Memory (LSTM) [9]. LSTM has been recently shown to be effective in learning a variable-length sequence-to-sequence mapping [5, 28].

Both question and answer words are represented with one-hot vector encoding (a binary vector with exactly one non-zero entry at the position indicating the index of the word in the vocabulary) and embedded in a lower dimensional space, using a jointly learnt latent linear embedding. In the training phase, we augment the question words sequence $\boldsymbol{q}$ with the corresponding ground truth answer words sequence $\boldsymbol{a}$, i.e. $\hat{\boldsymbol{q}} := [\boldsymbol{q}, \boldsymbol{a}]$. During the test time, in the prediction phase, at time step t, we augment $\boldsymbol{q}$ with previously predicted answer words $\hat{\boldsymbol{a}}_{1..t} := [\hat{\boldsymbol{a}}_1, \ldots, \hat{\boldsymbol{a}}_{t-1}]$, i.e. $\hat{\boldsymbol{q}}_t := [\boldsymbol{q}, \hat{\boldsymbol{a}}_{1..t}]$. This means the question $\boldsymbol{q}$ and the previous answers are encoded implicitly in the hidden states of the LSTM, while the latent hidden representation is learnt. We encode the image $\boldsymbol{x}$ using a CNN and provide it at every time step as input to the LSTM. We set the input $\boldsymbol{v}_t$ as a concatenation of $[\boldsymbol{x}, \hat{\boldsymbol{q}}_t]$.

As visualized in detail in Figure 3, the LSTM unit takes an input vector $\boldsymbol{v}_t$ at each time step t and predicts an output word $\boldsymbol{z_t}$ which is equal to its latent hidden state $\boldsymbol{h}_t$. As discussed above $\boldsymbol{z_t}$ is a linear embedding of the corresponding answer word $\boldsymbol{a_t}$. In contrast to a simple RNN unit the LSTM unit additionally maintains a memory cell $\boldsymbol{c}$. This allows to learn long-term dynamics more easily and significantly reduces the vanishing and exploding gradients problem [9]. More precisely, we use the LSTM unit as described in [36] and the *Caffe* implementation from [5]. With the *sigmoid* nonlinearity $\sigma : \mathbb{R} \mapsto [0,1]$, $\sigma(v) = (1+e^{-v})^{-1}$ and the *hyperbolic tangent* nonlinearity $\phi : \mathbb{R} \mapsto [-1,1]$, $\phi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} = 2\sigma(2v) - 1$, the LSTM updates for time step t given inputs $\boldsymbol{v}_t$, $\boldsymbol{h}_{t-1}$, and the memory cell $\boldsymbol{c}_{t-1}$ as follows:

$$\boldsymbol{i}_t = \sigma(W_{vi}\boldsymbol{v}_t + W_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{3}$$
$$\boldsymbol{f}_t = \sigma(W_{vf}\boldsymbol{v}_t + W_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{4}$$
$$\boldsymbol{o}_t = \sigma(W_{vo}\boldsymbol{v}_t + W_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{5}$$
$$\boldsymbol{g}_t = \phi(W_{vg}\boldsymbol{v}_t + W_{hg}\boldsymbol{h}_{t-1} + \boldsymbol{b}_g) \tag{6}$$
$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t \tag{7}$$
$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \phi(\boldsymbol{c}_t) \tag{8}$$

where $\odot$ denotes element-wise multiplication. All the weights W and biases b of the network are learnt jointly with the cross-entropy loss. Conceptually, as shown in

Figure 3, Equation 3 corresponds to the input gate, Equation 6 the input modulation gate, and Equation 4 the forget gate, which determines how much to keep from the previous memory c_{t-1} state. As Figures 1 and 2 suggest, all the output predictions that occur before the question mark are excluded from the loss computation, so that the model is penalized solely based on the predicted answer words.

Implementation We use default hyper-parameters of LSTM [5] and CNN [11]. All CNN models are first pre-trained on the ImageNet dataset [25], and next we randomly initialize and train the last layer together with the LSTM network on the task. We find this step crucial in obtaining good results. We have explored the use of a 2 layered LSTM model, but have consistently obtained worse performance. In a pilot study, we have found that *GoogleNet* architecture [11, 29] consistently outperforms the *AlexNet* architecture [11, 16] as a CNN model for our task and model.

4. Experiments

In this section we benchmark our method on a task of answering questions about images. We compare different variants of our proposed model to prior work in Section 4.1. In addition, in Section 4.2, we analyze how well questions can be answered without using the image in order to gain an understanding of biases in form of prior knowledge and common sense. We provide a new human baseline for this task. In Section 4.3 we discuss ambiguities in the question answering tasks and analyze them further by introducing metrics that are sensitive to these phenomena. In particular, the WUPS score [20] is extended to a consensus metric that considers multiple human answers. Additional results are available in the supplementary material and on the project webpage [1].

Experimental protocol We evaluate our approach on the DAQUAR dataset [20] which provides $12,468$ human question answer pairs on images of indoor scenes [26] and follow the same evaluation protocol by providing results on accuracy and the WUPS score at $\{0.9, 0.0\}$. We run experiments for the full dataset as well as their proposed reduced set that restricts the output space to only 37 object categories and uses 25 test images. In addition, we also evaluate the methods on different subsets of DAQUAR where only 1, 2, 3 or 4 word answers are present.

WUPS scores We base our experiments as well as the consensus metrics on WUPS scores [20]. The metric is a generalization of the accuracy measure that accounts for word-level ambiguities in the answer words. For instance 'carton' and 'box' can be associated with a similar concept,

[1] https://www.d2.mpi-inf.mpg.de/visual-turing-challenge

	Accuracy	WUPS @0.9	WUPS @0.0
Malinowski et al. [20]	7.86	11.86	38.79
Neural-Image-QA (ours)			
- multiple words	17.49	23.28	57.76
- single word	**19.43**	**25.28**	**62.00**
Human answers [20]	50.20	50.82	67.27
Language only (ours)			
- multiple words	17.06	22.30	56.53
- single word	17.15	22.80	58.42
Human answers, no images	7.34	13.17	35.56

Table 1. Results on DAQUAR, all classes, single reference, in %.

and hence models should not be strongly penalized for this type of mistakes. Formally:

$$\text{WUPS}(A,T) = \frac{1}{N}\sum_{i=1}^{N} \min\{ \prod_{a\in A^i} \max_{t\in T^i} \mu(a,t), \prod_{t\in T^i} \max_{a\in A^i} \mu(a,t)\}$$

To embrace the aforementioned ambiguities, [20] suggest using a thresholded taxonomy-based Wu-Palmer similarity [34] for μ. The smaller the threshold the more forgiving metric. As in [20], we report WUPS at two extremes, 0.0 and 0.9.

4.1. Evaluation of Neural-Image-QA

We start with the evaluation of our Neural-Image-QA on the full DAQUAR dataset in order to study different variants and training conditions. Afterwards we evaluate on the reduced DAQUAR for additional points of comparison to prior work.

Results on full DAQUAR Table 1 shows the results of our Neural-Image-QA method on the full set ("multiple words") with 653 images and 5673 question-answer pairs available at test time. In addition, we evaluate a variant that is trained to predict only a single word ("single word") as well as a variant that does not use visual features ("Language only"). In comparison to the prior work [20] (shown in the first row in Table 1), we observe strong improvements of over 9% points in accuracy and over 11% in the WUPS scores [second row in Table 1 that corresponds to "multiple words"]. Note that, we achieve this improvement despite the fact that the only published number available for the comparison on the full set uses ground truth object annotations [20] – which puts our method at a disadvantage. Further improvements are observed when we train only on a single word answer, which doubles the accuracy obtained

	Accuracy	WUPS @0.9	WUPS @0.0
Neural-Image-QA (ours)	21.67	27.99	65.11
Language only (ours)	19.13	25.16	61.51

Table 2. Results of the single word model on the one-word answers subset of DAQUAR, all classes, single reference, in %.

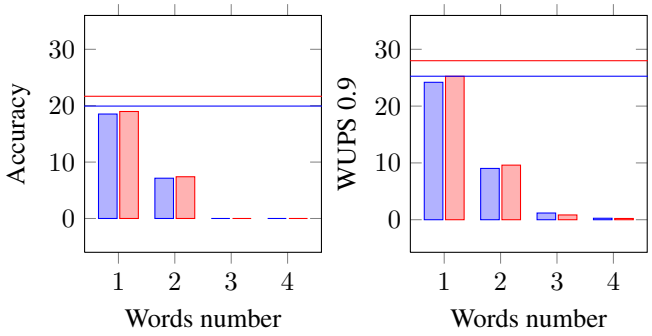

Figure 4. Language only (blue bar) and Neural-Image-QA (red bar) "multi word" models evaluated on different subsets of DAQUAR. We consider 1, 2, 3, 4 word subsets. The blue and red horizontal lines represent "single word" variants evaluated on the answers with exactly 1 word.

in prior work. We attribute this to a joint training of the language and visual representations and the dataset bias, where about 90% of the answers contain only a single word.

We further analyze this effect in Figure 4, where we show performance of our approach ("multiple words") in dependence on the number of words in the answer (truncated at 4 words due to the diminishing performance). The performance of the "single word" variants on the one-word subset are shown as horizontal lines. Although accuracy drops rapidly for longer answers, our model is capable of producing a significant number of correct two words answers. The "single word" variants have an edge on the single answers and benefit from the dataset bias towards these type of answers. Quantitative results of the "single word" model on the one-word answers subset of DAQUAR are shown in Table 2. While we have made substantial progress compared to prior work, there is still a 30% points margin to human accuracy and 25 in WUPS score ["Human answers" in Table 1].

Results on reduced DAQUAR In order to provide performance numbers that are comparable to the proposed Multi-World approach in [20], we also run our method on the reduced set with 37 object classes and only 25 images with 297 question-answer pairs at test time.

Table 3 shows that Neural-Image-QA also improves on the reduced DAQUAR set, achieving 34.68% Accuracy and 40.76% WUPS at 0.9 substantially outperforming [20] by

	Accuracy	WUPS @0.9	WUPS @0.0
Malinowski et al. [20]	12.73	18.10	51.47
Neural-Image-QA (ours)			
- multiple words	29.27	36.50	79.47
- single word	**34.68**	**40.76**	79.54
Language only (ours)			
- multiple words	32.32	38.39	80.05
- single word	31.65	38.35	**80.08**

Table 3. Results on reduced DAQUAR, single reference, with a reduced set of 37 object classes and 25 test images with 297 question-answer pairs, in %

21.95% Accuracy and 22.6 WUPS. Similarly to previous experiments, we achieve the best performance using the "single word" variant.

4.2. Answering questions without looking at images

In order to study how much information is already contained in questions, we train a version of our model that ignores the visual input. The results are shown in Table 1 and Table 3 under "Language only (ours)". The best "Language only" models with 17.15% and 32.32% compare very well in terms of accuracy to the best models that include vision. The latter achieve 19.43% and 34.68% on the full and reduced set respectively.

In order to further analyze this finding, we have collected a new human baseline "Human answer, no image", where we have asked participants to answer on the DAQUAR questions without looking at the images. It turns out that humans can guess the correct answer in 7.86% of the cases by exploiting prior knowledge and common sense. Interestingly, our best "language only" model outperforms the human baseline by over 9%. A substantial number of answers are plausible and resemble a form of common sense knowledge employed by humans to infer answers without having seen the image.

4.3. Human Consensus

We observe that in many cases there is an inter human agreement in the answers for a given image and question and this is also reflected by the human baseline performance on the question answering task of 50.20% ["Human answers" in Table 1]. We study and analyze this effect further by extending our dataset to multiple human reference answers in Section 4.3.1, and proposing a new measure – inspired by the work in psychology [4, 6, 23] – that handles disagreement in Section 4.3.2, as well as conducting additional experiments in Section 4.3.3.

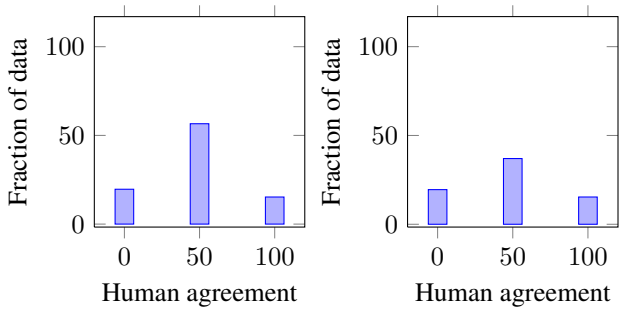

Figure 5. Study of inter human agreement. At x-axis: no consensus (0%), at least half consensus (50%), full consensus (100%). Results in %. Left: consensus on the whole data, right: consensus on the test data.

4.3.1 DAQUAR-Consensus

In order to study the effects of consensus in the question answering task, we have asked multiple participants to answer the same question of the DAQUAR dataset given the respective image. We follow the same scheme as in the original data collection effort, where the answer is a set of words or numbers. We do not impose any further restrictions on the answers. This extends the original data [20] to an average of 5 test answers per image and question. We refer to this dataset as DAQUAR-Consensus.

4.3.2 Consensus Measures

While we have to acknowledge inherent ambiguities in our task, we seek a metric that prefers an answer that is commonly seen as preferred. We make two proposals:

Average Consensus: We use our new annotation set that contains multiple answers per question in order to compute an expected score in the evaluation:

$$\frac{1}{NK}\sum_{i=1}^{N}\sum_{k=1}^{K}\min\{\prod_{a\in A^i}\max_{t\in T_k^i}\mu(a,t),\ \prod_{t\in T_k^i}\max_{a\in A^i}\mu(a,t)\} \tag{9}$$

where for the i-th question A^i is the answer generated by the architecture and T_k^i is the k-th possible human answer corresponding to the k-th interpretation of the question. Both answers A^i and T_k^i are sets of the words, and μ is a membership measure, for instance WUP [34]. We call this metric "Average Consensus Metric (ACM)" since, in the limits, as K approaches the total number of humans, we truly measure the inter human agreement of every question.

Min Consensus: The Average Consensus Metric puts more weights on more "mainstream" answers due to the summation over possible answers given by humans. In order to measure if the result was at least with one human in agreement, we propose a "Min Consensus Metric (MCM)" by replacing the averaging in Equation 9 with a max operator. We call such metric Min Consensus and suggest using both metrics in the benchmarks. We will make the implementation of both metrics publicly available.

$$\frac{1}{N}\sum_{i=1}^{N}\max_{k=1}^{K}\left(\min\{\prod_{a\in A^i}\max_{t\in T_k^i}\mu(a,t),\ \prod_{t\in T_k^i}\max_{a\in A^i}\mu(a,t)\}\right) \tag{10}$$

Intuitively, the max operator uses in evaluation a human answer that is the closest to the predicted one – which represents a minimal form of consensus.

	Accuracy	WUPS @0.9	WUPS @0.0
Subset: No agreement			
Language only (ours)			
- multiple words	8.86	12.46	38.89
- single word	8.50	12.05	40.94
Neural-Image-QA (ours)			
- multiple words	**10.31**	**13.39**	40.05
- single word	9.13	13.06	**43.48**
Subset: $\geq 50\%$ agreement			
Language only (ours)			
- multiple words	21.17	27.43	66.68
- single word	20.73	27.38	67.69
Neural-Image-QA (ours)			
- multiple words	20.45	27.71	67.30
- single word	**24.10**	**30.94**	**71.95**
Subset: Full Agreement			
Language only (ours)			
- multiple words	27.86	35.26	78.83
- single word	25.26	32.89	79.08
Neural-Image-QA (ours)			
- multiple words	22.85	33.29	78.56
- single word	**29.62**	**37.71**	**82.31**

Table 4. Results on DAQUAR, all classes, single reference in % (the subsets are chosen based on DAQUAR-Consensus).

4.3.3 Consensus results

Using the multiple reference answers in DAQUAR-Consensus we can show a more detailed analysis of inter human agreement. Figure 5 shows the fraction of the data where the answers agree between all available questions ("100"), at least 50% of the available questions and do not agree at all (no agreement - "0"). We observe that for the majority of the data, there is a partial agreement, but even full disagreement is possible. We split the dataset

	Accuracy	WUPS @0.9	WUPS @0.0
Average Consensus Metric			
Language only (ours)			
- multiple words	11.60	18.24	52.68
- single word	11.57	18.97	54.39
Neural-Image-QA (ours)			
- multiple words	11.31	18.62	53.21
- single word	**13.51**	**21.36**	**58.03**
Min Consensus Metric			
Language only (ours)			
- multiple words	22.14	29.43	66.88
- single word	22.56	30.93	69.82
Neural-Image-QA (ours)			
- multiple words	22.74	30.54	68.17
- single word	**26.53**	**34.87**	**74.51**

Table 5. Results on DAQUAR-Consensus, all classes, consensus in %.

into three parts according to the above criteria "No agreement", "$\geq 50\%$ agreement" and "Full agreement" and evaluate our models on these splits (Table 4 summarizes the results). On subsets with stronger agreement, we achieve substantial gains of up to 10% and 20% points in accuracy over the full set (Table 1) and the **Subset: No agreement** (Table 4), respectively. These splits can be seen as curated versions of DAQUAR, which allows studies with factored out ambiguities.

The aforementioned "Average Consensus Metric" generalizes the notion of the agreement, and encourages predictions of the most agreeable answers. On the other hand "Min Consensus Metric" has a desired effect of providing a more optimistic evaluation. Table 5 shows the application of both measures to our data and models.

Moreover, Table 6 shows that "MCM" applied to human answers at test time captures ambiguities in interpreting questions by improving the score of the human baseline from [20] (here, as opposed to Table 5, we exclude the original human answers from the measure). It also cooperates well with WUPS at 0.9, which takes word ambiguities into account, gaining an about 20% higher score.

4.4. Qualitative results

We show predicted answers of different variants of our architecture in Table 7, 8, and 9. We have chosen the examples to highlight differences between Neural-Image-QA and the "Language only". We use a "multiple words" approach only in Table 8, otherwise the "single word" model is shown. Despite some failure cases, "Language only" makes "reasonable guesses" like predicting that the largest object could be table or an object that could be found on the bed is either a pillow or doll.

	Accuracy	WUPS @0.9	WUPS @0.0
WUPS [20]	50.20	50.82	67.27
ACM (ours)	36.78	45.68	64.10
MCM (ours)	60.50	69.65	82.40

Table 6. Min and Average Consensus on human answers from DAQUAR, as reference sentence we use all answers in DAQUAR-Consensus which are not in DAQUAR, in %

4.5. Failure cases

While our method answers correctly on a large part of the challenge (e.g. $\approx$ 35 WUPS at 0.9 on "what color" and "how many" question subsets), spatial relations ($\approx$ 21 WUPS at 0.9) which account for a substantial part of DAQUAR remain challenging. Other errors involve questions with small objects, negations, and shapes (below 12 WUPS at 0.9). Too few training data points for the aforementioned cases may contribute to these mistakes.

Table 9 shows examples of failure cases that include (in order) strong occlusion, a possible answer not captured by our ground truth answers, and unusual instances (red toaster).

5. Conclusions

We have presented a neural architecture for answering natural language questions about images that contrasts with prior efforts based on semantic parsing and outperforms prior work by doubling performance on this challenging task. A variant of our model that does not use the image to answer the question performs only slightly worse and even outperforms a new human baseline that we have collected under the same condition. We conclude that our model has learnt biases and patterns that can be seen as forms of common sense and prior knowledge that humans use to accomplish this task. We observe that indoor scene statistics, spatial reasoning, and small objects are not well captured by the global CNN representation, but the true limitations of this representation can only be explored on larger datasets. We extended our existing DAQUAR dataset to DAQUAR-Consensus, which now provides multiple reference answers which allows to study inter-human agreement and consensus on the question answer task. We propose two new metrics: "Average Consensus", which takes into account human disagreement, and "Min Consensus" that captures disagreement in human question answering.

Acknowledgements. Marcus Rohrbach was supported by a fellowship within the FITweltweit-Program of the German Academic Exchange Service (DAAD).

			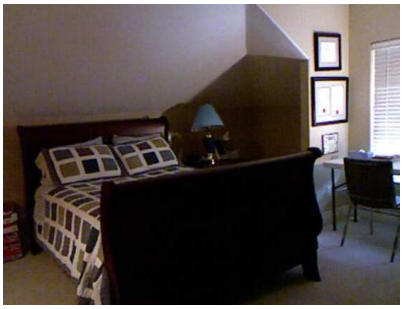
	What is on the right side of the cabinet?	How many drawers are there?	What is the largest object?
Neural-Image-QA:	bed	3	bed
Language only:	bed	6	table

Table 7. Examples of questions and answers. Correct predictions are colored in green, incorrect in red.

	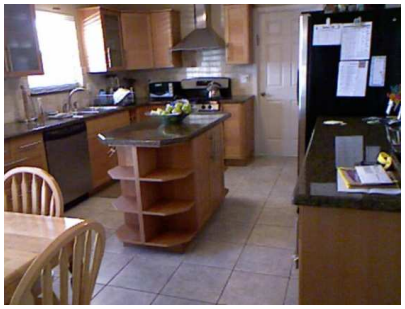		
	What is on the refrigerator?	What is the colour of the comforter?	What objects are found on the bed?
Neural-Image-QA:	magnet, paper	blue, white	bed sheets, pillow
Language only:	magnet, paper	blue, green, red, yellow	doll, pillow

Table 8. Examples of questions and answers with multiple words. Correct predictions are colored in green, incorrect in red.

			
	How many chairs are there?	What is the object fixed on the window?	Which item is red in colour?
Neural-Image-QA:	1	curtain	remote control
Language only:	4	curtain	clock
Ground truth answers:	2	handle	toaster

Table 9. Examples of questions and answers - failure cases.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. *arXiv:1505.00468*, 2015.

[2] J. Berant and P. Liang. Semantic parsing via paraphrasing. In *ACL*, 2014.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, D. Bahdanau, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[4] J. Cohen et al. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 1960.

[5] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[6] J. L. Fleiss and J. Cohen. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 1973.

[7] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015.

[8] D. Geman, S. Geman, N. Hallonquist, and L. Younes. Visual turing test for computer vision systems. *Proceedings of the National Academy of Sciences*, 2015.

[9] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.

[10] M. Iyyer, J. Boyd-Graber, L. Claudino, R. Socher, and H. D. III. A neural network for factoid question answering over paragraphs. In *EMNLP*, 2014.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014.

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[13] A. Karpathy, A. Joulin, and L. Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014.

[14] C. Kong, D. Lin, M. Bansal, R. Urtasun, and S. Fidler. What are you talking about? text-to-image coreference. In *CVPR*, 2014.

[15] J. Krishnamurthy and T. Kollar. Jointly learning to parse and perceive: Connecting natural language to the physical world. *TACL*, 2013.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[18] P. Liang, M. I. Jordan, and D. Klein. Learning dependency-based compositional semantics. *Computational Linguistics*, 2013.

[19] L. Ma, Z. Lu, and H. Li. Learning to answer questions from image using convolutional neural network. *arXiv:1506.00333*, 2015.

[20] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014.

[21] M. Malinowski and M. Fritz. Towards a visual turing challenge. In *Learning Semantics (NIPS workshop)*, 2014.

[22] C. Matuszek, N. Fitzgerald, L. Zettlemoyer, L. Bo, and D. Fox. A joint model of language and perception for grounded attribute learning. In *ICML*, 2012.

[23] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *ACL*, 2013.

[24] M. Ren, R. Kiros, and R. Zemel. Image question answering: A visual semantic embedding model and a new dataset. In *NIPS*, 2015.

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *arXiv:1409.0575*, 2014.

[26] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[28] I. Sutskever, O. Vinyals, and Q. V. V. Le. Sequence to sequence learning with neural networks. In *NIPS*. 2014.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv:1409.4842*, 2014.

[30] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence – video to text. In *ICCV*, 2015.

[31] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL*, 2015.

[32] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *arXiv:1411.4555*, 2014.

[33] J. Weston, S. Chopra, and A. Bordes. Memory networks. *arXiv:1410.3916*, 2014.

[34] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *ACL*, 1994.

[35] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank image generation and question answering. *arXiv:1506.00278*, 2015.

[36] W. Zaremba and I. Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

[37] C. L. Zitnick, D. Parikh, and L. Vanderwende. Learning the visual interpretation of sentences. In *ICCV*, 2013.