# Question Answering about Images using Visual Semantic Embeddings


## Abstract

This work aims to address the problem of image-based question-answering (QA) with new models and datasets. In our work, we propose to use recurrent neural networks and visual semantic embeddings without intermediate stages such as object detection and image segmentation. Our model performs 1.7 times better than the recently published results on the same dataset. Another main contribution is an automatic question generation algorithm that converts the currently available image description dataset into QA form, resulting in a 10 times bigger dataset with more even answer distribution.


## 1. Introduction

Combining image understanding and natural language interaction is one of the grand dream of artificial intelligence. We are interested in the problem of joinly learning image and text through a question-answering task. Recently, image caption generation (Kiros et al., 2014; Vinyals et al., 2014; Xu et al., 2015) has shown us feasible ways of jointly learning image and text to form higher level representations from models such as convolutional neural networks (CNNs) trained on object recognition and word embeddings trained on large scale corpus. Image QA involves an extra layer of interation between human and computers. Here the input is combined image and text and the model needs to output an answer that is targeted to the question asked. The model needs to pay attention to details of the image instead of describing it in a vague sense. The problem also combines many computer vision subproblems such as image labelling and object detection. Unlike traditional computer vision tasks, it is not obvious what kind of task the model needs to perform unless it understands the question.

In this paper we present our contribution to the problem: a generic end-to-end QA model using visual semantic embeddings to connect CNN and recurrent neural net (RNN), an automatic question generation algorithm that converts description sentences into questions, and a synthetic QA dataset which is generated using the algorithm. We will make the our synthetic COCO-QA dataset publicly available upon the release of the paper and we invite follow-up research in the community to contribute to this problem.


Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.


## 2. Problem Formulation

The input of the problem is a piece of image, a question of a sequence of words, and the output is an answer which is also a sequence of words. In this work, however, we assume that the answers are only single-word, which allows us to treat the problem as a classification problem. The assumption is made based on the fact that 98.3% of the examples in the recently released dataset (Malinowski & Fritz, 2014a) are single-word. This also makes the evaluation of the models easier and more robust.

## 3. Related Work

In 2014, (Malinowski & Fritz, 2014a) released a dataset with images and question-answer pairs. It is called DAtaset for QUestion Answering on Real-world images (DAQUAR) . All images are from the NYU depth v2 dataset (Nathan Silberman & Fergus, 2012), and are taken from indoor scenes. Human segmentations, image depth values, and object labellings are available in the dataset. The QA data has two sets of configuration: the 37-class and the 894-class dataset, differed by the number of object classes appearing in the questions. There are mainly three types of questions in this dataset: object type, object color, and number of objects. Some questions are easy but many questions are very hard to answer even to humans. Figure 1 shows some examples of easy and hard questions. Since DAQUAR is the only publicly available image-based QA dataset, it is one of our benchmarks to evaluate our models.

Together with the release of DAQUAR dataset, (Malinowski & Fritz, 2014b) presented an approach which combines semantic parsing and image segmentation. In the natural language part of their work, they used a semantic parser (Liang et al., 2013) to convert sentences into latent logical forms. They obtained the multiple segmentations of the image by sampling the uncertainty of the segmentation algorithm. Their model is based on a Bayesian formulation that every logical form and image segmentation has

certain probability. They also converted every image segmentation to a bag of predicates. To make their algorithm scalable, they chose to sample from the nearest neighbors in the training set according to the similarity of the predicates.

As there are not much existing work done in the field of image question answering, their attempt have large room for improvement. First, although they are handling a number of spatial relations, a human-defined possible set of predicates are very dataset-specific. To obtain the predicates, their algorithm also depends on a good image segmentation algorithm and image depth information. Second, before asking any of the questions, their model needs to compute all possible spatial relations in the training images, so even for a small dataset like 1500 images there could be 4 millions predicates in the worst case (Malinowski & Fritz, 2014b). Even though the model searches from the nearest neighbors of the test images, it could still be an expensive operation in larger datasets. Lastly the accuracy of their model is not very strong. We will show later that some simple baselines will perform better in terms of plain accuracy.

## 4. Proposed Methodology

The methodology presented here is two-folded. On the model side we applied recurrent neural networks and visual-semantic embeddings on this task, and on the dataset side we proposed new ways of synthesizing QA pairs from currently available image description dataset.

### 4.1. Models

In recent years, recurrent neural networks (RNNs) had successful applications in the field of natural language processing (NLP). Long short-term memory (LSTM) is a type of RNN which is easier to train than other regular RNNs because of its linear error propagation and multiplicative gatings. There has been increasing interests in using LSTM as encoders and decoders on sentence level. Our model builds directly on top of the LSTM sentence model and is called the "CNN-LSTM" model. It treats the image as one word of the question. We borrowed this idea of treating the image as a word from caption generation work done by (Vinyals et al., 2014). The difference with caption generation is that here we only output the answer at the last time step.

1. We used the last hidden layer of the Oxford Conv Net (Simonyan & Zisserman, 2014) as our visual embeddings. The CNN part of our model is kept frozen during training.

2. We experimented with several different word embedding models: randomly initialized embedding, dataset-specific skip-gram embedding and general-purpose skip-gram embedding model (Mikolov et al., 2013). The word embeddings can either be frozen or dynamic.

3. We then treated the image as if it is the first word of the sentence. Similar to (Frome et al., 2013), we used a linear or affine transformation to map 4096 dimension image feature vectors to 300 or 500 dimension that matches the dimension of the word embeddings.

4. We can optionally treat the image as the last word of the question as well through a different transformation matrix.

5. We can optionally add a backward direction LSTM, which get the same content but in a backward sequential fashion.

6. The LSTM(s) output(s) to a softmax layer at the last timestep to classify answers.

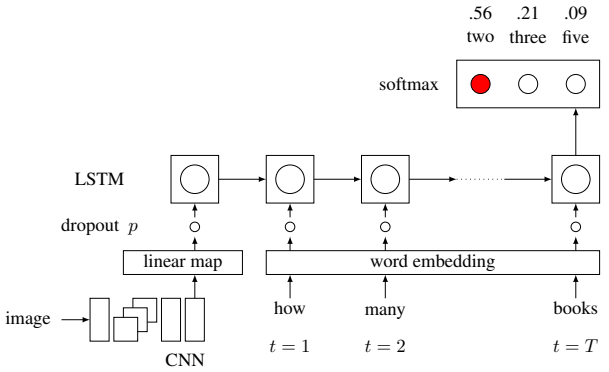



*Figure 2.* CNN-LSTM model

### 4.2. Datasets

The currently available DAQUAR dataset only has around 1500 images with 7000 images on 37 common object classes, which might be not enough for training large complex models. Another problem with the current dataset is that the guessing the mode can get very good accuracy. So by creating another dataset, we can make much larger number of QA pairs and more even answer distribution. While collecting human generated QA pairs could be a possible way, we propose to automatically convert descriptions into QA forms. As a starting point we used Microsoft-COCO dataset (Lin et al., 2014) but it can extend to any other image description dataset such as Flickr dataset (Hodosh et al., 2013), SBU dataset (Ordonez et al., 2011), or even directly from the internet. Another advantage of using image description is that they are generated by humans in the first place. So many objects mentioned in the description

Q2355: what is the colour of roll of tissue paper ?
Ground truth: white
Easy because toilet tissue paper is always white

Q1466: what is on the night stand ?
Ground truth: paper
Object is too small to focus

Q2010: what is to the right of the table ?
Ground truth: sink
Too many cluttered objects

*Figure 1.* Variety of Difficulty Levels in DAQUAR

are easier to notice than the ones in original QAs and synthetic QAs generated from ground truth labellings. It allows the model to rely more on common sense and rough image understanding without any logical reasoning. Lastly the conversion process preserves the language variability in the original description and will result in more human-like questions than questions on image labellings.

Question generation is still an open-ended topic. We are not trying to solve a linguistic problem but just to create a usable image question answering dataset. We prefer undergenerating questions rather than overgenerating. The reason is that currently available image description datasets are large in their sizes, so the quality of questions us more important.

#### 4.2.1. Common Strategies

1. Compound sentences to simple sentences
   Here we only consider a simple case that is when two sentences are joined together with a conjunctive words. We split the orginial sentences into two independent sentences. For example, "There is a cat and the cat is running." will be split as "There is a cat." and "The cat is running."

2. Indefinite determiners to definite determiners.
   Asking questions on a specific instance of the subject requires changing the determiner into definite form "the". For example, "**A** boy is playing baseball." will have "the" instead of "a" in its question form: "What is **the** boy playing?".

3. Wh-movement constraints
   For English language, questions tend to start with interrogative words such as "what". The algorithm needs to move the verb as well as the "wh-" constituent to the front of the sentence. In this work we consider the following two simple constraints:

   (a) A-over-A principle
   The A-over-A principle restricts the movement of a wh-word inside an NP (Chomsky, 1973). For example, "I am talking to John and **Bill**" cannot be transformed into "*__Who__ am I talking to John and" because "Bill" is an noun phrase (NP) that is under another NP "John and Bill".

   (b) Clauses
   Our algorithm does not move any wh-word that is contained in a clause constituent. This rule can be further fine tuned in the future.

#### 4.2.2. Pre-Processing

We used Stanford parser (Klein & Manning, 2003) to obtain syntatic structure of the original image description.

#### 4.2.3. Object-Type Questions

First, we consider asking an object using "what". This involves replacing the actual object with a "what" in the sentence, and then transforming the sentence structure so that the "what" appears in the front of the sentence. The entire algorithm has the following stages:

1. Split long sentences into simple sentences
2. Change indefinite determiners to definite determiners.
3. Traverse the sentence and identify potential answers and replace with "what". During the traversal of object-type question generation, we ignore all the prepositional phrase (PP) consituents because nouns inside prepositions like "of" and "with" are rarely meaningful answers.
4. Perform wh-movement

Figure 3 illustrates these procedures with tree diagrams. In order to identify a possible answer word, we used WordNet

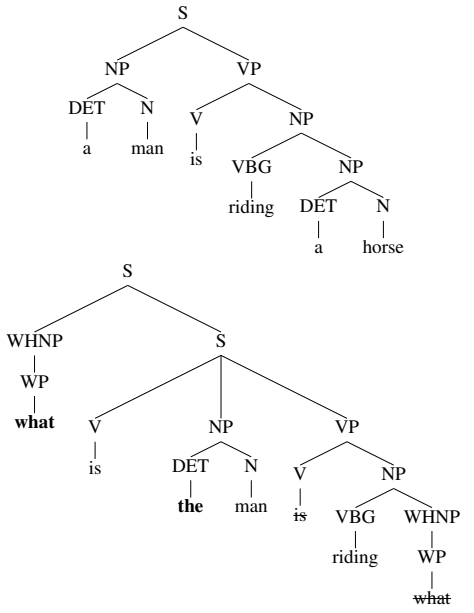



*Figure 3.* Example: "A man is riding a **horse**" => "**What** is **the** man riding?"

(Fellbaum, 1998) and NLTK software package (Bird, 2006) to get noun categories.

#### 4.2.4. Number-Type Questions

To generate "how many" types of questions, we follow a similar procedure as the previous algorithm, except for a different way to identify potential answers. This time, we need to extract numbers from original sentences. Splitting compound sentences, changing determiners, and wh-movement parts remain the same.

#### 4.2.5. Color-Type Questions

Color-type questions are much easier to generate. It only requires locating the colour adjective and the noun which the adjective attaches to. Then it simply forms a sentence "What is the colour of the object" with the "object" replaced by the actual noun.

#### 4.2.6. Location-Type Questions

Similar to generating object-type question, except that now the answer traversal will only search within PP contituents with preposition "in". We also added rules to filter out clothings so that the answers will mostly be locations, scenes, or large objects that contain smaller objects.

#### 4.2.7. Post-Processing

We will show in our experiment results that mode-guessing actually works unexpectedly well in DAQUAR dataset. One of our design requirement of the new dataset is to avoid too common answers. To achieve this goal, we applied a heuristic to reject the answers that appear too often in generated dataset. First, all QA pairs are shuffled to mitigate dependence between neighboring pairs. We formulate the rejection process as a Bernoulli random process. The probability of enrolling next QA pair $(q, a)$ is:

$$p(q,a) = \begin{cases} 1 & \text{if count}(a) \leq U \\ \exp\left(-\frac{count(a)-U}{2U}\right) & \text{otherwise} \end{cases} \quad (1)$$

where $count(a)$ denotes the current number of enrolled QA pairs that have $a$ as the ground truth answer. After this QA selection process we obtain more uniform distribution of all possible answers.

## 5. Experimental Results

### 5.1. Dataset

Table 5.1 summarizes the difference between DAQUAR and COCO-QA. COCO-QA has much more data to support the training of more complex models on a larger number of possible answer classes. Another important note is that since we applied QA pair selection process, mode-guessing performs very poorly on COCO-QA; however, since all the answers can be found in the original image descriptions, COCO-QA questions are actually easier to answer from a human point of view. This encourages the model to exploit salient object relations instead of exhaustively search from all possible relations.

*Table 1.* General statistics of DAQUAR and COCO-QA

| Dataset | DAQUAR-37 | COCO-QA |
|---|---|---|
| Images | 795+654 | 80K+40K |
| Questions | 3825+3284 | 83K+39K |
| Answers | 63 | 410 |
| GUESS | 0.1885 | 0.0665 |

Table 5.1 gives a break-down view of all four types of questions present in the COCO-QA dataset with train/test split information.

### 5.2. Proposed Model Details

We trained two versions of the architectures that are found to work the best. The first model is CNN and LSTM with a weight matrix in the middle and we call it "CNN-LSTM

*Table 2.* COCO-QA Question Type Break-Down

| CATEGORY | TRAIN | % | TEST | % |
|---|---|---|---|---|
| OBJECT | 58352 | 69.96% | 27497 | 71.13% |
| NUMBER | 6551 | 7.85% | 2575 | 6.66% |
| COLOR | 13337 | 15.99% | 6047 | 15.64% |
| LOCATION | 5164 | 6.19% | 2541 | 6.57% |
| TOTAL | 83404 | 100.00% | 38660 | 100.00% |

model" in our tables and figures. The second model has two image feature input at the start and the end of the sentence with different linear transformations, and additionally it has two LSTMs going from the forward and backward direction. Both LSTMs output to the softmax layer at the last timestep. We call the second model "2-CNN-LSTM model" in our tables and figures.

### 5.3. Baselines

To evaluate the effectiveness of our models, we designed a few baselines.

#### 5.3.1. GUESS MODEL

One very simple baseline is to predict the mode based on question type. For example, if the question contains "how many" then the model will output "two." This baseline actually works unexpectedly well in DAQUAR dataset.

#### 5.3.2. BLIND-BOW MODEL

To avoid over-interpretation of the results, we designed blind-model baselines which are given only the questions without the images. It is interesting to see how image features are useful in answering the questions. One of the simplest blind model is to simply sum all the word vectors of the question into a bag-of-word (BOW) vector. We then send the sentence features into a softmax layer. Optionally we added a tanh hidden layer before the softmax layer to allow more interaction between image features and word features.

#### 5.3.3. BLIND-LSTM MODEL

Another blind model we experimented is to input the question words into the LSTM alone. We could compare it with BLIND-BOW to see how does LSTM contribute to a better language understanding.

#### 5.3.4. CNN-BOW MODEL

Similar to BLIND-BOW model but replaced the feature layer with bag-of-word sentence features concatenated with image features from Oxford Net last hidden layer (4096 dimension) after a linear transformation (to 300 or 500 dimension).

### 5.4. Performance Metrics

To evaluate the model, we used the plain answer accuracy as well as the Wu-Palmer similarity (WUPS) measure (Wu & Palmer, 1994; Malinowski & Fritz, 2014b). The WUPS calculates the similarity between two words based on their longest common subsequence in the taxonomy tree. The similarity function takes in a threshold parameter. If the similarity between two words is less than the threshold then zero score will be given to the candidate answer. It reduces to plain accuracy when the threshold equals to 1.0. Following (Malinowski & Fritz, 2014b), we measure all the models in terms of plain accuracy and WUPS at 0.9 threshold.

Table 3 summarizes the learning results on DAQUAR dataset. Here we compare our results with (Malinowski & Fritz, 2014b)'s model. It should be noted that their results is for the entire dataset whereas our result is on 98.3% of the original dataset with single word answers.

*Table 3.* DAQUAR Results

| | ACC. | WUPS 0.9 |
|---|---|---|
| 2-CNN-LSTM | **0.3578** | **0.3602** |
| CNN-LSTM | 0.3441 | 0.3464 |
| BLIND-LSTM | 0.3273 | 0.3294 |
| BLIND-BOW | 0.3267 | 0.3289 |
| GUESS | 0.1824 | 0.1671 |
| MULTI-WORLD | 0.1273 | 0.1810 |
| HUMAN | 0.6027 | 0.6104 |

*Table 4.* COCO-QA Results

| | ACC. | WUPS 0.9 |
|---|---|---|
| 2-CNN-LSTM | **0.5161** | **0.5244** |
| CNN-LSTM | 0.5073 | 0.5153 |
| CNN-BOW | 0.4490 | 0.4593 |
| BLIND-LSTM | 0.3516 | 0.3592 |
| BLIND-BOW | 0.3262 | 0.3337 |
| GUESS | 0.0665 | 0.0851 |

*Table 5.* COCO-QA Accuracy Per Category Break-Down

| | OBJECT | NUMBER | COLOR | LOCATION |
|---|---|---|---|---|
| 2-CNN-LSTM | **0.5386** | 0.4534 | **0.4786** | 0.4258 |
| CNN-LSTM | 0.5321 | **0.4678** | 0.4439 | **0.4294** |
| CNN-BOW | 0.4718 | 0.3773 | 0.4130 | 0.3609 |
| BLIND-LSTM | 0.3459 | 0.4383 | 0.3334 | 0.3680 |
| BLIND-BOW | 0.3201 | 0.3339 | 0.3434 | 0.3436 |
| GUESS | 0.0211 | 0.3584 | 0.1387 | 0.0893 |

## 6. Discussion

From the above results we demonstrated our models is doing a reasonable job in the image question answering classification problem. It outperforms the baselines and the existing approach in terms of answer accuracy. It is amazing to see that the blind model is in fact very strong on DAQUAR dataset, even comparable to CNN-LSTM models. We speculate that it is likely that the ImageNet image examples is very different from the indoor scene images which are mostly furnitures. So therefore the CNN-LSTM cannot really make use of the image features unless the question is asking about the largest object, i.e. differentiating between sofas, beds, and tables. However, the CNN-LSTM model outperforms the blind model by a large margin in COCO-QA dataset. There are three possible reasons behind. First, the objects in MS-COCO dataset resemble the ones in ImageNet more; second, the images have less number of objects whereas the indoor scenes have many cluttered objects in one image; and third, COCO-QA has more data to train complex models.

There are many interesting examples but due to the space limit we can only show a few in Figure 4, 5, 6. The full result is available to view at (url will be ready upon the release of the paper). For some of the examples, we specifically tested extra questions (the ones have "a" in the question ID) to avoid over-interpretating the questions that CNN-LSTM accidentally got correct. The parentheses in the figures represent the confidence score given by the softmax layer of the models. "DQ" denotes questions from DAQUAR dataset and "CQ" denote questions from COCO-QA dataset.

### 6.1. Model Selection

We did not find that using different word embedding has a significant impact on the final classification results. The best model for DAQUAR is using a randomly initialized embedding of 300 dimensions whereas the best model trained for COCO-QA is using a problem-specific skip-gram embedding initialize. We find that normalizing the CNN hidden layer image features before send to linear/affine transformation helps the training become smoother because the image features will blend with the word embedding space with similar statistics. The bidirectional LSTM model can further boost the result by a little.

### 6.2. Object

As the original CNN was trained for the ImageNet (Russakovsky et al., 2014) challenge, the CNN-LSTM benefited largely from the object recognition ability. For some simple object pictures in COCO-QA dataset, the CNN-LSTM and CNN-BOW can easily get the correct answer just from the image features. However, the challenging part is to consider spatial relations between multiple objects and probably focus on details of the image. Some qualitative results in Figure 4 shows that the CNN-LSTM only does a moderately acceptable job on it. Sometimes it fails to make correct decision but keep output the most salient object or sometimes the blind model can equally guess the most probable objects based on the question alone (e.g. chairs should be around the dinning table).

### 6.3. Counting

In DAQUAR, we cannot observe any advantage in counting of the CNN-LSTM model compared to other baselines. In COCO-QA there is some observable counting ability in very clean image with single object type. The model can sometimes count up to five or six. However, as shown in Figure 5, the ability is fairly weak as it does not count correctly when different object types are present. As the CNN-LSTM model only wins the blind one by 3% on counting, there will be very large space for improvement in the counting task and in fact it could be a separate computer vision problem by its own.

### 6.4. Color

In COCO-QA there is significant difference between CNN-LSTM model and BLIND-LSTM model on color-type questions. We further discovered that the model not only can recognize the dominant color of the image but can sometimes associating different color to different object, as shown in Figure 6. However, the model is still not very robust and fails on a number of easy examples.

### 6.5. Limitations and Future Work

Image question answering is a fairly new research topic, and the approaches we present here has a number of limitations. First the model is just an answer classifier. Ideally we would like to permit longer answers which will involve some sophisticated text generation model or structured output learning. Secondly, our question generation algorithm

also assumes that all answers are single word, and the implementation of the algorithm is heavily dependent on the question type. Also, the algorithm is only applicable to English language at this time. In this paper, the question generation was only serving for our models so the above limitations were not big concerns. Lastly, our approach is hard to interpret why the model output a certain answer. By comparing the model to some baselines we can roughly infer but humans are still prone to over-interpretion of the results. Visual attentions might be future direction for us which could also help explaining the model output.

## 7. Conclusion

In this paper, we consider the image QA problem and present our CNN-LSTM model by combining CNN and LSTMs with visual-semantic embeddings. As the currently available dataset is not large enough, we designed an algorithm that helps us collect large scale image QA dataset from image descriptions. Our model shows a reasonable understanding of the question and some coarse image understanding, but it is still very naive in many situations. Moreover, our question generation algorithm is extensible to many image description datasets and can be automated without the involvement of human labor. We hope that the creation of the new dataset can encourage more data-driven approaches to this problem in the future.

## Acknowledgments

## References


Bird, Steven. NLTK: the natural language toolkit. In ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006, 2006.

Chomsky, Noam. Conditions on Transformations. Academic Press, New York, 1973.

Fellbaum, Christiane (ed.). WordNet An Electronic Lexical Database. The MIT Press, Cambridge, MA ; London, May 1998. ISBN 978-0-262-06197-1.

Frome, Andrea, Corrado, Gregory S., Shlens, Jonathon, Bengio, Samy, Dean, Jeffrey, Ranzato, Marc'Aurelio, and Mikolov, Tomas. Devise: A deep visual-semantic embedding model. In Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013., pp. 2121–2129, 2013.

Hodosh, Micah, Young, Peter, and Hockenmaier, Julia. Framing image description as a ranking task: Data, models and evaluation metrics. J. Artif. Intell. Res. (JAIR), 47:853–899, 2013.

Kiros, Ryan, Salakhutdinov, Ruslan, and Zemel, Richard S. Unifying visual-semantic embeddings with multimodal neural language models. CoRR, abs/1411.2539, 2014. URL http://arxiv.org/abs/1411.2539.

Klein, Dan and Manning, Christopher D. Accurate unlexicalized parsing. In In proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 423–430, 2003.

Liang, Percy, Jordan, Michael I., and Klein, Dan. Learning dependency-based compositional semantics. Computational Linguistics, 39(2):389–446, 2013.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C. Lawrence. Microsoft COCO: common objects in context. In Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, pp. 740–755, 2014.

Malinowski, Mateusz and Fritz, Mario. Towards a visual turing challenge. CoRR, abs/1410.8027, 2014a. URL http://arxiv.org/abs/1410.8027.

Malinowski, Mateusz and Fritz, Mario. A multi-world approach to question answering about real-world scenes based on uncertain input. In Neural Information Processing Systems (NIPS'14), 2014b. URL http://arxiv.org/abs/1410.0210.

Mikolov, Tomas, Chen, Kai, Corrado, Greg, and Dean, Jeffrey. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013. URL http://arxiv.org/abs/1301.3781.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli and Fergus, Rob. Indoor segmentation and support inference from rgbd images. In ECCV, 2012.

Ordonez, Vicente, Kulkarni, Girish, and Berg, Tamara L. Im2text: Describing images using 1 million captioned photographs. In Neural Information Processing Systems (NIPS), 2011.

Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, 2014.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Vinyals, Oriol, Toshev, Alexander, Bengio, Samy, and Erhan, Dumitru. Show and tell: A neural image caption generator. CoRR, abs/1411.4555, 2014. URL http://arxiv.org/abs/1411.4555.

Wu, Zhibiao and Palmer, Martha. Verb semantics and lexical selection. In In Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pp. 133–138, 1994.

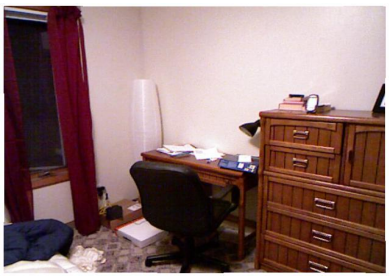
Xu, Kelvin, Ba, Jimmy, Kiros, Ryan, Cho, Kyunghyun, Courville, Aaron, Salakhutdinov, Ruslan, Zemel, Richard S., and Bengio, Yoshua. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. ArXiv e-prints, February 2015.

CQ5429: what do two women hold with a picture on it ?
Ground truth: cake
2-CNN-LSTM: cake (0.5611)
CNN-BOW: laptop (0.1443)
BLIND-LSTM: umbrellas (0.1567)
BLIND-BOW: phones (0.1447)

CQ24952: what is the black and white cat wearing ?
Ground truth: hat
2-CNN-LSTM: hat (0.6349)
BLIND-LSTM: tie (0.5821)

CQ25218: where are the ripe bananas sitting ?
Ground truth: basket
2-CNN-LSTM: basket (0.4965)
BLIND-LSTM: bowl (0.6415)
CQ25218a: what are in the basket ?
Ground truth: bananas
2-CNN-LSTM: bananas (0.6443)
BLIND-LSTM: bears (0.0956)

DQ585: what is the object on the chair?
Ground truth: pillow
2-CNN-LSTM: pillow (0.6475)
BLIND-LSTM: clothes (0.3973)
DQ585a: where is the pillow found?
Ground truth: chair
2-CNN-LSTM: chair (0.1735)
BLIND-LSTM: cabinet (0.7913)



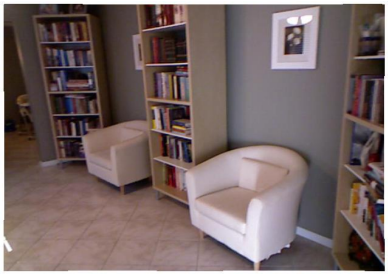
DQ2136: what is right of table?
Ground truth: shelves
2-CNN-LSTM: shelves (0.2780)
BLIND-LSTM: shelves (0.2000)
DQ2136a: what is in front of table?
Ground truth: chair
2-CNN-LSTM: chair (0.3104)
BLIND-LSTM: chair (0.3661)
Sometimes the blind model can infer indoor object relations without looking at the image.


CQ2007: what is next to the cup?
Ground truth: banana
2-CNN-LSTM: banana (0.3560)
BLIND-LSTM: sandwich (0.0647)
CQ2007a: what is sitting next to a banana?
Ground truth: cup
2-CNN-LSTM: banana (0.4539)
BLIND-LSTM: cat (0.0538)
In this example the CNN-LSTM model fails to output different objects based on spatial relations.

*Figure 4.* Qualitative Evaluation on Object-Type Questions


CQ6805: how many uncooked doughnuts sit on the baking tray?
Ground truth: six
2-CNN-LSTM: six (0.2779)
CNN-BOW: twelve (0.2342)
BLIND-LSTM: four (0.2345)


CQ32912: how many bottles of beer are there?
Ground truth: three
2-CNN-LSTM: three (0.4153)
CNN-BOW: two (0.4849)
CQ32912a: how many bananas are there?
Ground truth: two
2-CNN-LSTM: three (0.1935)
CNN-BOW: two (0.5307)



DQ1520: how many shelves are there ?
Ground truth: three
2-CNN-LSTM: two (0.4801)
BLIND-LSTM: two (0.2060)
DQ1520a: how many sofas are there ?
Ground truth: two
2-CNN-LSTM: two (0.6173)
BLIND-LSTM: two (0.5378)
In the last two examples the model does not know how to count with different object types.

*Figure 5.* Qualitative Evaluation on Number-Type Questions

DQ2989: what is the colour of the sofa?
Ground truth: red
2-CNN-LSTM: red (0.2152)
BLIND-LSTM: brown (0.2061)
DQ2989a: what is the colour of the table?
Ground truth: white
2-CNN-LSTM: white (0.4057)
BLIND-LSTM: white (0.2751)

CQ6200: what is the color of the cone?
Ground truth: yellow
2-CNN-LSTM: yellow (0.4250)
CNN-BOW: white (0.5507)
BLIND-LSTM: orange (0.4473)
CQ6200a: what is the color of the bear ?
Ground truth: white
2-CNN-LSTM: white (0.6000)
CNN-BOW: white (0.5518)
BLIND-LSTM: brown (0.5518)

CQ6058: what is the color of the coat?
Ground truth: yellow
2-CNN-LSTM: yellow (0.2942)
BLIND-LSTM: black (0.2456)
CQ6058a: what is the color of the umbrella?
Ground truth: red
2-CNN-LSTM: yellow (0.4755)
BLIND-LSTM: purple (0.2243)

*Figure 6.* Qualitative Evaluation on Color-Type Questions