Answer-Type Prediction for Visual Question Answering

Kushal Kafle and Christopher Kanan*
Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology
{kk6055, kanan}@rit.edu

Abstract

Recently, algorithms for object recognition and related tasks have become sufficiently proficient that new vision tasks can now be pursued. In this paper, we build a system capable of answering open-ended text-based questions about images, which is known as Visual Question Answering (VQA). Our approach's key insight is that we can predict the form of the answer from the question. We formulate our solution in a Bayesian framework. When our approach is combined with a discriminative model, the combined model achieves state-of-the-art results on four benchmark datasets for open-ended VQA: DAQUAR, COCO-QA, The VQA Dataset, and Visual7W.

1. Introduction

Using deep convolutional neural networks (CNNs) [9], object recognition systems now rival the abilities of humans on benchmarks such as the ImageNet Large Scale Visual Recognition Challenge [16]. Implicitly, these systems ask the question, "What is the dominant object in this image?" Similarly, recent captioning methods [2, 21, 19, 7, 3] are attempting to answer the implicit question "What are the main entities and activities in the image?" However, often there are numerous questions that can be answered about an image beyond recognizing the dominant object or activity. A natural way to address this is to build a system that is given an image and a text-based question, and then it outputs a text-based answer. This is known as the open-ended Visual Question Answering (VQA) problem [1]. VQA requires merging computer vision with natural language processing (NLP). It is especially challenging because models for VQA need to be implicitly capable of object recognition, object detection, attribute recognition, and more. Until now, the main obstacle to pursuing the VQA problem was a lack of datasets containing image-question-answer pairs; however, five publicly available datasets for VQA became available in 2015: DAQUAR [11], COCO-QA [15], The VQA Dataset [1], FM-IQA, and Visual7W [23]. In this paper, we study multiple VQA models and evaluate them on four of these datasets. Our main contribution is to observe that when answering a question, it is generally possible to predict the form the answer will take. For example, for the question "Is it raining?" a valid answer will be either "yes" or "no." The answer will never be "green" or "10." However, existing models do not have this kind of reasoning explicitly built into them. Incorporating information predicted about the answer can also potentially improve a model's internal representation to handle the question.

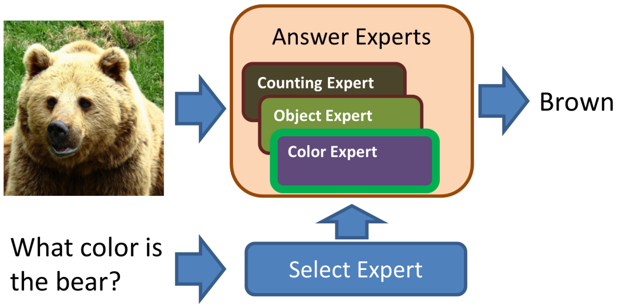


Figure 1: In the open-ended VQA problem, an algorithm is given an image and a question, and it must output a string containing the answer. We obtain state-of-the-art results on multiple VQA datasets by adopting a Bayesian approach that incorporates information about the form the answer should take. In this example, the system is given an image of a bear and it is asked about the color of the bear. Our method explicitly infers that this is a "color" question and uses that information in its predictive process.

We first describe a Bayesian framework for VQA that incorporates answer-type prediction. We then show that we can use text-based features to predict with greater than 99% accuracy the form the answer will take for all of the datasets. We then evaluate and compare our model against methods from the literature and a discriminative model trained using

*Corresponding author.

DAQUAR: What colors do the stools around the table have ?
Ground Truth: blue, white
DAQUAR: What is leaning against the wall on the left side of the white cabinet?
Ground Truth: Ladder

COCO-VQA: Do the horses legs look strong enough to support its body?
Ground Truth: Yes
COCO-QA: What is the color of the horses?
Ground Truth: Brown

COCO-VQA: What direction are the giraffes facing?
Ground Truth: to right; away;left; 3 towards camera; away from camera; forward; west; toward building
COCO-QA: What just standing around an old stable as their picture is taken?
Ground Truth: Giraffes

Figure 2: Images and their corresponding question-answer pairs from the COCO-VQA, COCO-QA, and DAQUAR datasets. DAQUAR is generated using human annotators and it has unambiguous questions, but it has solely indoor images that tend to have many small objects. COCO-VQA is generated by human annotators and has a wide variety of questions, but some questions have ambiguous or subjective answers. COCO-QA is generated using an automated algorithm that produces one word answers, but some questions are grammatically incorrect.

the same features. Another contribution is the use of skip-thought vectors [8], which have not previously been used for VQA. Skip-thought vectors are a recently developed technique for encoding sentences into vectors in a manner that preserves salient sentence information. We are also the first to evaluate our models on each of the publicly available datasets for VQA, and we provide a critical analysis of each dataset's strengths and weaknesses. A demonstration of a simplified version of our algorithm can be found at http://askimage.org.

2. Datasets for Visual Question Answering

Five publicly available datasets for open-ended VQA have been recently released: COCO-QA, The VQA Dataset, DAQUAR, Visual7W, and FM-IQA. COCO-QA, FM-IQA, and Visual7W contain question-answer pairs for images from the Microsoft Common Objects in Context (COCO) dataset [10]. Over half of The VQA Dataset is also comprised of COCO images, and we refer to this portion of the dataset as COCO-VQA. COCO consists of 328,000 images, 91 common object categories, and over 2 million labeled instances. Each image also has 5 captions. In the following subsections, we discuss how question-answer pairs for these datasets were created, and we also briefly describe their strengths and shortcomings.

2.1. DAQUAR

DAQUAR (The DAtaset for QUestion Answering on Real-world images) [11] is a collection of question-answer pairs for the NYU Depth V2 dataset [17]. Example image is shown in Fig. 2. The dataset is available in two configurations. DAQUAR-FULL consists of 6795 (train) and 5673 (test) question-answer pairs. DAQUAR-37 has 37 object categories, and 3825 (train) and 297 (test) question-answer pairs. Following others [11, 15], we only use the portion of these datasets that have single-word answers.

One of the limitations of DAQUAR is that it contains exclusively indoor scenes, which constrains the variety of questions available. Moreover, the images often contain significant clutter and numerous small objects, making some questions very difficult to answer. Even people have significant difficulty with this dataset, with humans only achieving 50.20% accuracy on DAQUAR-FULL [12].

2.2. COCO-QA

COCO-QA [15] contains 78,736 QA pairs for training and 38,948 pairs for testing. QA pairs were generated automatically from the COCO captions using an question generating algorithm. COCO-QA has 430 unique, one word answers. All questions belong to one of four categories: object (69.84%), number (7.47%), color (16.59%), and location (6.10%). Because the questions are derived automatically, many questions in COCO-QA are awkwardly posed or grammatically incorrect (see Fig. 2).

2.3. COCO-VQA

COCO-VQA is the subset of The VQA Dataset that has been created from real-world images drawn from COCO [1]. The remainder of The VQA Dataset contains synthetic images, which we do not discuss further here. COCO-VQA is further split into multiple-choice and open-

Table 1: A comparison of publicly available VQA datasets with open-ended answers. Note that the numbers for COCO-VQA come from the subset of The VQA Dataset in which the answers are open-ended and the images come from COCO.

	DAQUAR	COCO-QA	COCO-VQA	Visual-7W
Total Images	795 Train 654 Test	82783 Train 40504 Test	82783 Train 40504 Validation 81434 Test	47300
QA Pairs	10620 Train 5970 Test	78736 Train 38948 Test	248349 Train 121512 Validation 244302 Test	327939
Distinct Answers	968	430	145172	25553 (84163 choices)
Longest Question	25 Words	24 Words	32 Words	24 Words
Longest Answer	7 Word List	1 Word	17 Words	20
Answer Format	Words, Lists	Single Word only	Words, Phrases	Phrases, Sentences
Image Source	NYUDv2	COCO	COCO	COCO
QA Generation	Human+Algorithms	Algorithms	Human	Human
Answer Types	Color, Object, Number	Color, Object, Number and Location	Unlimited (Roughly into,“yes/no,” “number,” and “other”)	7-W(what, where, how, when, who, why,which) Questions

ended answers. Our results are solely on the open-ended portion of COCO-VQA.

COCO-VQA consists of 614,163 human generated free-form questions with 6,141,630 human responses (10 responses per question). Question-answer pairs are often complex and rich, and answering many of them requires object recognition or activity recognition as well as contextual and knowledge based reasoning.

While diversity of questions in COCO-VQA is impressive, the unconstrained method used to collect questions also produces some difficult QA pairs. These can be roughly categorized into the following: 1) Questions requiring subjective judgment or opinion to answer; 2) Questions with answers that are not deducible from image content alone; and 3) Questions with no clear answer. Fig. 2 shows examples from COCO-VQA.

2.4. Visual7W

The Visual7W dataset [23] contains 327,939 questions for 47,300 COCO images. Visual7W consists of multiple choice questions-answer pairs consisting of six types of ’W’ questions *(what, when, who, why, where, how)*. It also has *’which’* type questions where the answer can be in visual form. In addition to the question-answer pairs, Visual7W has annotated bounding boxes that refer to the objects referred to in question-answer pairs. In our experiments we use the open-ended *telling* portion of the dataset, in which the multiple-choice options and all *’which’* questions are removed.

2.5. FM-IQA

FM-IQA [4] is another dataset created from COCO. It has questions and answers in Chinese, with English translations. A limited subset of English translations have been publicly released (102 “yes/no” questions and 33464 “what” questions). Because most FM-IQA answers are sentences, it makes it difficult to automatically evaluate VQA algorithms on this dataset, and in [4], the authors used human judges to evaluate how well a method performed. For these reasons, we do not use FM-IQA in our experiments.

3. Related Work

Although VQA is a new problem, sophisticated algorithms for VQA are already being deployed. Most existing papers on VQA have used Long-Short-Term-Memory (LSTM) neural networks. [4], [12], [1], and [15] all used LSTM networks to encode the question and combined the question encoding with image features from a deep convolutional neural network (CNN). Each of these took a slightly different approach, and we summarize them below.

In [1], the authors’ best model on COCO-VQA was an LSTM model with a 1000-node softmax output layer, which generated answers for the top-1000 most frequent answers. Their LSTM model used a one-hot encoding of question words and CNN features from a pre-trained network. A linear transformation mapped the CNN features to the same dimensionality of the question words. These were then combined using the element-wise (Hadamard) product, and then fed into a MLP network.

In [15], a similar approach was taken, with the main difference being that they fed CNN features to the LSTM as the first "word," followed by vectors encoding each word of the sentence, and then finally the last word was the CNN features once more. In a variant of this approach, [12] sequentially gave their LSTM network concatenated CNN and word features at every time step.

In [4], separate LSTMs were used for the question and answer, but they had a shared word embedding layer, and CNN image features are fused at the end of the LSTM. Their model was able to output more than one-word answers or lists, and it could generate coherent sentences.

As an alternative to LSTM networks, in [11] the authors created a Bayesian framework for VQA. They used semantic segmentation to get information about the objects present in an image, such as their categories and spatial locations. Then, their Bayesian framework calculated the probability of each answer given the semantic segmentation image features and the question.

Unlike us, none of these methods explicitly incorporated information about the answer-type. Also, instead of using an LSTM network, we encode questions using skip-thought vectors (see section 6.4).

4. Evaluation of VQA Systems

The most straightforward measure used to evaluate VQA systems is accuracy, *i.e.*, the system must output exactly the same answer as the human annotator. However, difficulties arise in two situations. First, many questions have multiple valid answers, *e.g.*, "What is on the table?" might have "Mandarin Orange," "orange," or "fruit" as valid answers. Using one-to-one matching will penalize a model if it does not output exactly the same answer as the human annotator.

One way to handle this problem is to use the Wu-Palmer Similarity (WUPS) index [20], which is used to evaluate VQA systems in [15], [11] and [12]. WUPS ranges between 0 through 1, where 1.0 is perfect match between semantic meaning of two words being compared. This measure can be used to relax the stringent requirement of accuracy measure which unnecessarily penalizes semantically similar answers, *e.g.*, "Mandarin Orange" and "Orange" have a WUPS Score of 0.9286 which indicates high similarity between the words. WUPS calculates similarity between two specific word senses but each word can have multiple senses. In this regard, [11] suggest using a metric that considers similarity between all possible combinations between a set of word senses produced from two words being compared and returns maximum similarity between them.

However, standard WUPS can assign relatively high scores to semantically unrelated words, *e.g.*, "Dog" and "Orange" have a score of 0.58. To mitigate this, [11] suggested applying a threshold to the WUPS score, in which a score below the threshold is multiplicatively scaled by some factor, and they suggested using a threshold of 0.9,which suggests a high correlation, with scaling factor of 0.1. Along with accuracy, this modified version of WUPS is the standard used on both COCO-QA and DAQUAR. This does not completely resolve the semantic ambiguity problem, because WUPS can assign a high score to words with diametrically opposite meanings, *e.g.*, "White" and "Black" have WUPS score of 0.9, because they belong to the same general category.

Another way to address ambiguous answers is to get answers from multiple human annotators for each question. This is the approach taken in The VQA Dataset [1] as well as in a version of DAQUAR dubbed DAQUAR-Consensus [12]. This information can be used to place lower emphasis on questions where humans also tend to disagree. Furthermore, multiple human answers to a question enables us to study performance as a function of human agreement, and both [1] and [12] found that models performed worse on questions where human agreement is low. In [12], a measure called consensus was proposed, which assigns a score based on inter-human agreement on a question's answer. Similarly, [1] proposed an accuracy metric in which a predicted answer is deemed correct if three or more people gave the same answer. Specifically, they proposed to evaluate the accuracy of a question's answer using

$$\text{Accuracy}_{\text{VQA}} = \min(\frac{n}{3}, 1), \tag{1}$$

where n is the number of people that gave the predicted answer. With this measure, it will be impossible for a system to reach 100% accuracy on COCO-VQA because there are many examples in which less than three annotators agreed on the answer. This occurs for over 59% of "Why" questions in the COCO-VQA training data. Moreover, over 13% of yes/no questions in COCO-VQA's training data have *both* yes and no repeated three times, causing opposite answers to both count as correct. This often occurs with subjective questions, *e.g.*, "Would you eat this?"

Despite the problems described above, we used the standard evaluation metrics for each dataset in order to allow direct comparison with existing results *i.e.*, plain accuracy and WUPS scores (code from [12]) on COCO-QA and DAQUAR, the and accuracy measure from Equation 1 on COCO-VQA via the official evaluation server.

5. Predicting the Answer-Type

Our method requires each question to be assigned a type during training. The way we do this differs for each dataset.

DAQUAR does not have explicitly defined answer categories. We created three categories by looking at the answers: Number, Color, and Other. We assigned all answers that were numbers to the number category, all answers that were one of the 10 canonical colors (black, white, blue, brown, gray, green, orange, purple, red) to the color category, all other answers were assigned to the other category.

COCO-QA has four explicitly defined answer categories: Object, Color, Counting (Number), and Location. We did not change them. For COCO-VQA, 'Yes/No', 'Number', and 'Other' types are explicitly defined (denoted DT for default types). Besides using the default types, we also used an extended set of types that we constructed with heuristics (denoted ET for extended types). We subdivided the 'Number' category into 'counting' and 'other numbers' by looking at whether the question began with 'how many.' Answers that were 14 common colors (black, white, blue, brown, gray, green, orange, purple, red, silver, gold, tan and pink) were assigned to the color category, if the question contained the word 'color'. 'COCO objects' was assigned if the answer was one of the object categories defined in COCO. Finally, all questions terminating in 'playing' or 'doing' were assigned the type 'activity.' All remaining questions were grouped under the 'others' type category.

Across all of the datasets, we were able to use our skip-thought representation with logistic regression to infer the answer type for questions with over 99.7% accuracy on validation data.

6. Models for VQA

6.1. A New Bayesian Model for VQA

We formulate the VQA problem in a Bayesian framework. Let $\mathbf{x}$ be a column vector containing image features and $\mathbf{q}$ be a column vector containing question features. Given a question and an image, our model estimates the probability of a particular answer k and question-type c as $P\left(A=k, T=c|\mathbf{x},\mathbf{q}\right)$. Using Bayes' rule and the chain rule for probabilities, this can be expressed as

$$P\left(A=k, T=c|\mathbf{x},\mathbf{q}\right) = \frac{P\left(\mathbf{x}|A=k,T=c,\mathbf{q}\right)P\left(A=k|T=c,\mathbf{q}\right)P\left(T=c|\mathbf{q}\right)}{P\left(\mathbf{x}|\mathbf{q}\right)},$$

where $P\left(\mathbf{x}|A=k,T=c,\mathbf{q}\right)$ is the probability of the image features given the answer, answer-type, and question, $P\left(A=k|T=c,\mathbf{q}\right)$ is the probability of the answer given the answer-type, and question, $P\left(T=c|\mathbf{q}\right)$ is the probability of the answer-type given the question, and $P\left(\mathbf{x}|\mathbf{q}\right)$ is the probability of the image features given the question. To obtain the answer to a question about an image, we can simply marginalize over all of the answer types, *i.e.*,

$$P\left(A=k|\mathbf{x},\mathbf{q}\right) = \sum_{c\in T} P\left(A=k,T=c|\mathbf{x},\mathbf{q}\right).$$

While it is possible to train all aspects of the model jointly using a maximum likelihood solution, we chose to use simple models that are trained individually for each distribution. This makes training simple and fast. We model $P\left(A=k|T=c,\mathbf{q}\right)$ and $P\left(T=c|\mathbf{q}\right)$ using logistic regression classifiers. Because $P\left(\mathbf{x}|\mathbf{q}\right)$ does not influence the prediction, it can be disregarded.

We model each $P\left(\mathbf{x}|A=k,T=c,\mathbf{q}\right)$ with a conditional multivariate Gaussian, *i.e.*,

$$P\left(\mathbf{x}|A=k,T=c,\mathbf{q}\right) = \mathcal{N}\left(\mathbf{x}|\bar{\boldsymbol{\mu}}_{k,c,\boldsymbol{q}}, \bar{\boldsymbol{\Sigma}}_{k,c}\right).$$

This approach shares similarities with attention, in that it directly models that the image features that should be paid attention to should depend on the question. It is related to Quadratic Discriminant Analysis (QDA) [5]; however, in standard QDA the Gaussians are not conditional on additional features, unlike our approach.

The conditional mean and covariance for each Gaussian is computed as follows. Let the sample mean and covariance for the training data with answer k and answer-type c, in which the image features $\boldsymbol{x}$ are concatenated with the question features $\boldsymbol{q}$, be $\boldsymbol{\mu}_{k,c} = \left[\begin{array}{cc}\boldsymbol{\mu}_{k,c,\mathbf{x}} & \boldsymbol{\mu}_{k,c,\mathbf{q}}\end{array}\right]^T$ and

$$\boldsymbol{\Sigma}_{k,c} = \left[\begin{array}{cc}\boldsymbol{\Sigma}_{k,c,1,1} & \boldsymbol{\Sigma}_{k,c,1,2}\\ \boldsymbol{\Sigma}_{k,c,2,1} & \boldsymbol{\Sigma}_{k,c,2,2}\end{array}\right].$$

Then, the mean of the Gaussian given $\boldsymbol{q}$ is

$$\bar{\boldsymbol{\mu}}_{k,c,\mathbf{q}} = \boldsymbol{\mu}_{k,c,\mathbf{x}} + \boldsymbol{\Sigma}_{k,c,1,2}\boldsymbol{\Sigma}^{-1}_{k,c,2,2}\left(\mathbf{q}-\boldsymbol{\mu}_{k,c,\mathbf{q}}\right)$$

and the covariance will be

$$\bar{\boldsymbol{\Sigma}}_{k,c} = \boldsymbol{\Sigma}_{k,c,1,1} - \boldsymbol{\Sigma}_{k,c,1,2}\boldsymbol{\Sigma}^{-1}_{k,c,2,2}\boldsymbol{\Sigma}_{k,c,2,1}.$$

Note that the new mean for the image features depends on the question features, but the new covariance does not.

Because we have limited training data for some answer and answer-type combinations, estimating $\boldsymbol{\Sigma}_{k,c}$ accurately is difficult and the estimate should be regularized to ensure we can invert the covariance sub-matrices. To remedy this, we use a locally smoothed solution combined with shrinkage to estimate $\boldsymbol{\Sigma}_{k,c}$ [22], which is given by

$$\boldsymbol{\Sigma}_{k,c} = \frac{n_{k,c}\left(1-\beta\right)\Sigma'_{k,c} + \frac{1}{\kappa}\beta \sum\limits_{j\in KNN(k,c)} n_{j,c}\boldsymbol{\Sigma}'_{j,c}}{n_{k,c}\left(1-\beta\right) + \frac{1}{\kappa}\beta \sum\limits_{j\in KNN(k,c)} n_{j,c}} + \epsilon\mathbf{I}$$

where $\boldsymbol{\Sigma}'_{k,c}$ is the sample covariance matrix for the data with answer k and answer-type c and $n_{k,c}$ is the corresponding number of samples, $\mathbf{I}$ is the identity matrix, ϵ and β are scalar regularization parameters, and KNN$(\cdot)$ denotes the categories of the same type that have means with the smallest κ Euclidean distances to $\boldsymbol{\mu}_{k,c}$. We used $\kappa = 10$, $\epsilon = 0.01$, and $\beta = 0.4$ in all of our experiments.

In preliminary experiments, we also tried modeling $P\left(\mathbf{x}|A=k,T=c,\mathbf{q}\right)$ using conditionalized kernel density estimation with Gaussian kernels and using a conditionalized Gaussian mixture model, but in both cases performance was significantly worse on validation data than simply using a single Gaussian per answer.

6.2. Baseline Models

In addition to comparing to the models in the literature, we also tested five baseline models ourselves.

1. IMAGE: A logistic regression classifier trained with image features. It knows nothing about the question.

2. IMAGE+TYPE: For each answer-type in the dataset, we train a logistic regression classifier. We use our answer-type prediction model to select among the logistic regression classifiers for a given question, but the classifier does not have access to detailed question information. A similar approach was used in [15], where they used a question-type oracle to select among image feature classifiers on COCO-QA.

3. QUESTION: A logistic regression classifier trained only with the question features.

4. IMAGE+QUESTION: A logistic regression classifier trained with the image features concatenated to the question features.

5. MLP: A multi-layer perceptron network with a softmax output layer, with the image and question features as input. MLP is a 4-layer neural network with 6000 units on the first layer, 4000 for the second, 2000 for the third, and finally a softmax output layer with units equal to the number of categories. All hidden layers used rectified linear units. To regularize the network, drop-out of 0.3 was used with the hidden layers as well the input data layer.

6.3. Hybrid Model

Hybrid models that combine generative and discriminative classifiers can achieve a lower error rate than either alone [14]. Motivated by this, we created a hybrid approach that multiplicatively combines the two models, *i.e.*,

$$P_H\left(A=k|\mathbf{x},\mathbf{q}\right) \propto P_B\left(A=k|\mathbf{x},\mathbf{q}\right) P_D\left(A=k|\mathbf{x},\mathbf{q}\right)^{\alpha},$$

where $P_B\left(A=k|\mathbf{x},\mathbf{q}\right)$ is our Bayesian model, $P_D\left(A=k|\mathbf{x},\mathbf{q}\right)$ is IMAGE+QUESTION as described earlier, and α is a parameter that weights the distributions appropriately. This kind of weighting is a common approach to combining classifiers that were independently trained [18]. For DAQUAR and COCO-QA, we do five-fold cross-validation on the training data to find a good value for α, and for COCO-VQA α is tuned using the validation data. In both cases, we searched for α over $0.0, 0.1, 0.2, \ldots, 6$. This approach is labeled HYBRID. Additionally, for COCO-VQA, we used a variation where we combined our Bayesian model with MLP (HYBRID-MLP).

6.4. Question and Image Feature Representations

We use skip-thought vectors [8] to encode the text of a question into a vector $\boldsymbol{q}$, which have not previously been used for VQA. Skip-thought vectors are trained in an encoder-decoder framework, in which both the encoder and decoder are recurrent neural networks with gated recurrent units. The model is trained to encode a sentence, and it uses that encoding to reconstruct the previous and next sentence. They can therefore be trained in an unsupervised manner from corpuses of text. After training, the output of the encoder can be used as a rich feature vector, which was shown to achieve excellent performance on a variety of NLP classification tasks when used with a linear classifier. We use the 4800-dimensional combine-skip model from [8], which is a concatenation of uni-skip and bi-skip models. Each skip-thought vector is normalized to unit length.

For our image features $\boldsymbol{x}$, we used ResNet [6], with $448 \times 448 \times 3$ images. The features were taken from the last hidden layer after the ReLU, and then pooled across all spatial locations. These features were normalized to unit length. For the Bayesian model, we reduced the dimensionality of the CNN features using linear discriminant analysis to $K-1$ dimensions, where K is the number of possible answers.

7. Experiments

7.1. DAQUAR

Results on DAQUAR are shown in Table 2. For accuracy, both DAQUAR-FULL, QUESTION, IMAGE+QUESTION, BAYESIAN, and HYBRID all outperformed the prior state-of-the-art, with HYBRID performing best. A similar trend occurred for DAQUAR-37, with the exception of IMAGE+QUESTION. In both cases, we observe that QUESTION alone exceeds the previous state-of-the-art, and it performs extremely well compared to IMAGE alone. On DAQUAR-FULL, QUESTION achieves only slightly lower accuracy than the best performing HYBRID method. This may be because we and others used off-the-shelf CNN features that were tuned to recognize objects on ImageNet. These features tend to do a significant amount of spatial pooling, but DAQUAR questions are often about small objects in the image.

7.2. COCO-QA

Results on COCO-QA are shown in Table 2. For both accuracy and WUPS, HYBRID performed best, even though there was a gap in the performance of BAYESIAN and IMAGE+QUESTION. This suggests that the models are complementary, and they are making different mistakes. This did not occur with DAQUAR, and it may be because we have a lot more training data per answer on average for

Table 2: Results on DAQUAR-FULL, DAQUAR-37, and COCO-QA. All results on DAQUAR are for one-word answers.

	DAQUAR-FULL			**DAQUAR-37**			**COCO-QA**		
	Acc. (%)	WUPS 0.9	WUPS 0.0	Acc. (%)	WUPS 0.9	WUPS 0.0	Acc. (%)	WUPS 0.9	WUPS 0.0
MULTI-WORLD [11]	7.86	11.86	38.79	12.73	18.10	51.47	-	-	-
ASK-NEURON [12]	21.67	27.99	65.11	34.68	40.76	79.54	-	-	-
TORONTO-FULL [15]	-	-	-	36.94	48.15	82.68	57.84	67.90	89.52
IMAGE	6.19	11.31	45.83	7.93	13.13	54.38	34.36	46.63	72.58
QUESTION	25.57	31.49	67.09	39.66	44.19	82.19	39.24	50.11	83.42
IMAGE+TYPE	13.36	20.28	61.37	17.59	24.51	75.61	48.31	63.16	87.37
IMAGE+QUESTION	26.83	32.86	66.86	38.28	43.83	82.45	62.27	72.36	90.99
MLP	24.05	29.96	63.61	41.72	47.00	83.27	60.84	71.03	90.65
BAYESIAN	28.39	34.19	**67.48**	43.79	48.42	84.31	59.02	69.38	90.12
HYBRID	**28.96**	**34.74**	67.33	**45.17**	**49.74**	**85.13**	**63.18**	**73.14**	**91.32**

COCO-QA than we have for DAQUAR. We investigate this further in Section 7.5.

7.3. COCO-VQA

There are two subsets of COCO-VQA that are used for evaluation: Test-Dev and Test-Standard. The ground truth of both subsets is held by the creators of COCO-VQA, and it is necessary to upload predicted answers to their server for evaluation. Test-Dev is intended for development purposes, and Test-Standard is used to compare state-of-the-art methods. Researchers are currently only allowed to submit five results on Test-Standard and only one result file per day. We benchmark all of our methods on Test-Dev, and we benchmark the best performing of these on Test-Standard.

In our experiments on DAQUAR and COCO-QA, we trained our model to answer all answers in the training data; however, the number of possible answers is far greater for COCO-VQA (see Table 1). For COCO-VQA, we selected the most repeated answer for each question, and of these we only used top 1000 most common answers. This covers 82.67% of the answers in train and validation sets [1]. We did not use the remaining training data. Our results on COCO-VQA are given in Table 3. HYBRID methods performed well, with HYBRID-MLP using extended answer-types performing best on Test-Dev.

7.4. Visual7W

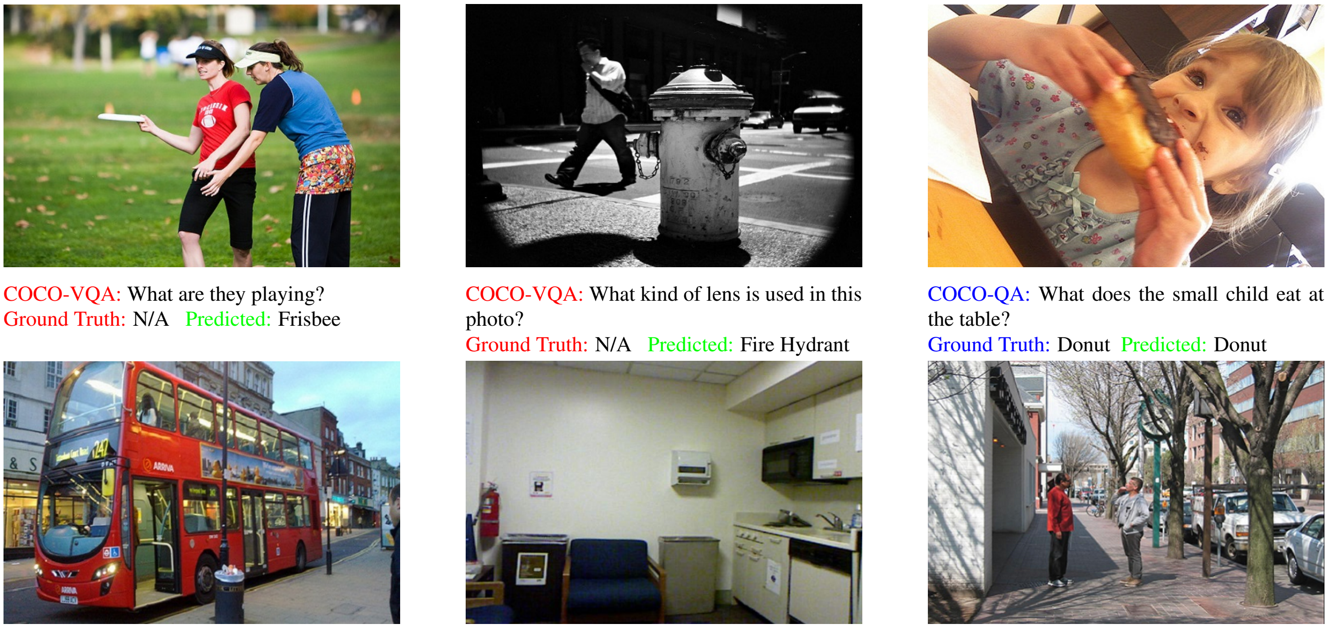
Visual7W results are shown in Table 4. Following [23], we show both top-1 accuracy and top-5 accuracy. For our experiments, the model was trained only with answers that occurred at least 20 times (536 total categories). HYBRID worked best, and it exceeded the prior state-of-the-art [23].

Table 3: Results on COCO-VQA, which were computed by uploading our models' predictions to the server run by the dataset's creators [1]. We compare against the best results in [1]. Key: IMG=IMAGE, QUES=QUESTION, BAYES=BAYESIAN, HYB=HYBRID, QTYPE=QUESTION TYPE DT=Default types, ET=Extended types.

	All	Yes/No	Number	Other
		Test Development		
LSTM Q+I [1]	53.74	78.94	35.24	36.42
IMG	29.59	70.65	0.38	1.16
QUES	49.56	77.36	35.49	29.02
IMG+QTYPE-DT	36.02	69.53	36.03	7.44
IMG+QTYPE-ET	44.74	69.49	34.76	25.88
IMG+QUES	54.92	76.92	35.77	40.46
BAYES-DT	53.49	77.00	35.13	37.57
BAYES-ET	54.58	77.58	35.03	39.66
HYB-DT	55.68	77.25	36.29	41.65
HYB-ET	56.00	77.21	36.10	42.38
MLP	58.65	79.93	36.80	45.42
HYB-MLP-DT	59.30	80.26	37.03	46.37
HYB-MLP-ET	**59.57**	**80.47**	**37.50**	**46.72**
		Test Standard		
LSTM Q+I [1]	54.06	79.01	35.55	36.80
HYB-MLP-ET	**60.06**	**80.34**	**37.82**	**47.56**

7.5. Bayesian vs. Discriminative

Generative models have been reported to outperform discriminative models when the amount of training data is low [13]. To investigate if this was the case here, we studied the difference in performance between our BAYESIAN and IMAGE+QUESTION models on answers with a different number of training examples on COCO-QA. We computed the median and mean number of examples for the training answers in which the Bayesian model performed better than the discriminative model and vice versa. For the



COCO-VQA: What are they playing?
Ground Truth: N/A Predicted: Frisbee

COCO-VQA: What kind of lens is used in this photo?
Ground Truth: N/A Predicted: Fire Hydrant

COCO-QA: What does the small child eat at the table?
Ground Truth: Donut Predicted: Donut

COCO-QA: What does the red , two level bus with it ; s open on the street and passengers inside the bus?
Ground Truth: Doors Predicted: Bus

DAQUAR: What is on the left side of the fire extinguisher and on the right side of the chair?
Ground Truth: Table Predicted: Table

Visual7W: 2. What color is the sidewalk
Ground Truth: Gray Predicted: Gray
Visual7W: 1. Where are the men talking?
GT: Sidewalk Predicted: In the street

Figure 3: Examples of correctly and incorrectly answered questions from each dataset using HYBRID for all datasets, except COCO-VQA where HYBRID-MLP-ET is shown. Because we do not have the answers for COCO-VQA's test datasets, we chose examples that subjectively looked correct or wrong to us.

answers in which the Bayesian model performed better, the median was 66 and the mean was 163.7, and for discriminative the median was 90 and the mean was 298.9. This is consistent with earlier observations [13], although it is somewhat surprising because the covariance matrices used in our Bayesian model require a significant amount of data to accurately estimate.

7.6. Does Answer-Type Prediction Help Accuracy?

Our proposed model directly incorporates answer-type prediction, but how useful is it? We studied this on DAQUAR-FULL, DAQUAR-37, and COCO-QA by doing experiments with a variant of our Bayesian model that did not incorporate explicit answer-type prediction. For DAQUAR-FULL and DAQUAR-37, we achieved similar performance with and without answer-type prediction (less than 0.5% difference in both cases). However, for COCO-QA we did find a meaningful improvement in performance with answer-type prediction, improving accuracy from 57.33% without answer-types to 59.02%. We also found that expanding the number of answer types improved accuracy on COCO-VQA.

Table 4: Top-1 accuracy and top-5 accuracy on Visual7W.

	Acc. (%)	**Top-5 Acc. (%)**
LSTM (Q+I) [23]	18.8	41.3
IMAGE	3.76	12.25
QUESTION	17.17	36.90
IMAGE+TYPE	8.39	25.70
IMAGE+QUESTION	22.07	43.34
MLP	20.76	42.73
BAYESIAN	19.23	39.83
HYBRID	**22.29**	**43.58**

8. Conclusion

In this paper, we proposed a Bayesian model for VQA that incorporates answer-type prediction, and we found that when it was combined with a discriminative model it achieved excellent results on four VQA datasets. Our Bayesian model is related to QDA, but we modified it to have a visual feature representation that is conditioned on the question features. We pioneered the use of skip-thought vectors for VQA, and we critically reviewed evaluation measures and datasets for open-ended VQA.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual question answering. In *ICCV*, 2015. 1, 2, 3, 4, 7

[2] 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. 1

[3] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 1

[4] 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. 3, 4

[5] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: data mining, inference and prediction. *New York: Springer-Verlag*, 1(8):371–406, 2001. 5

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[7] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 1

[8] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler. Skip-thought vectors. In *NIPS*, 2015. 2, 6

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014. 2

[11] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 1, 2, 4, 7

[12] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 2, 3, 4, 7

[13] A. Ng and A. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, 2002. 7, 8

[14] R. Raina, Y. Shen, A. Mccallum, and A. Y. Ng. Classification with hybrid generative/discriminative models. In *NIPS*, 2003. 6

[15] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015. 1, 2, 3, 4, 6, 7

[16] 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. In *IJCV*, 2015. 1

[17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 2

[18] A. Torralba, A. Oliva, M. S. Castelhano, and J. M. Henderson. Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. *Psychological review*, 113(4):766, 2006. 6

[19] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 1

[20] Z. Wu and M. Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994. 4

[21] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 1

[22] X.-Y. Zhang and C.-L. Liu. Locally smoothed modified quadratic discriminant function. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2013. 5

[23] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 1, 3, 7, 8