Visual Question Answering : How well do we learn ?

Bhargav Mangipudi
UIUC, Urbana, IL
mangipu2

Pramod Srinivasan
UIUC, Urbana, IL
psrnvsn2

Vishaal Mohan
UIUC, Urbana, IL
vmohan9

Abstract

Image-based question answering is a research area that is currently getting a lot of attention. The task is to answer natural language questions based on a given image. We build an end-to-end system using multiple deep-learning algorithms for this purpose and match the baseline accuracies for some cases (41.44% for Multilayer Perceptron, 46.69% for Heuristics-based method over MLP and 38.8% for an LSTM based model). Moreover, we conduct a thorough analysis of whether the accuracies that we report actually indicate how well the model understands the image/image features. As part of this analysis we consider questions that our model performs well on and manually curate some queries for the same information as in the original question to test if the performance we obtain is legit. We found that the accuracies that are reported by various algorithms are not a very good indicators of the amount of information we learn about the image.

1 Introduction

Visual Question Answering is an active interdisciplinary research area of computer vision, Natural Language Processing (NLP) and Machine Learning. Given an image and an image related Natural language learning question, VQA answers the question in a natural language sentence.

The VQA problem could potentially be of great importance to many applications such as image retrieval, blind person navigation and early child education. The recently released VQA data set (Antol et al., 2015) came with strong baselines based on CNN features combined with LSTM models. The questions are about specific information about the image and the answers require both common sense knowledge (Gao et al., 2015) and visual understanding (Ren et al., 2015).

In this project, we are only addressing the 'Multiple Choice' aspect of the problem. Specifically, our model output is one or more words that answer the question based on a given image.

Incidentally, this challenging task is still in its infancy, as almost all the work has been done in the last two years. Just after the conference CVPR'16 deadlines there have been several papers on arXiv proposing neural network architectures for VQA (Shih et al., 2015). As part of this project, we have initially set out to compare and contrast various models in terms of the accuracy. However the major focus of project is to analyze how meaningful these results are by accounting for the variations in accuracies, thereby ascertaining whether a given architecture may not normally be sufficient to fully exploit the relationship of the vision part and the question understanding part.

The rest of our report is structured as follows: in Section 2, we provide a short description of the datasets. In Section 3, we give an overview of the models before a detailed explanation of the experiments and analysis that followed in Section 4. We explain related work in Section 5, and provide conclusions and future work in Section 6.

2 Data Set description

MSCOCO-VQA is the recently released VQA data set which contains natural-language questions about images. This data set contains 369,861 questions and 3,698,610 ground truth answers based on 123,287 MSCOCO images. These questions and answers are sentence-based and open-ended. The training and testing split follows MSCOCO-VQA official split. Specifically, we use 82,783 images for training and 40,504 validation images for testing.

The initial phase of the project involved a sur-

vey of the recent work as well as selection of an ideal publicly available data set. We chose the VQA data set which is based on the images from the MSCOCO data set. There are about 120k images and all the 360k questions based on these images are human generated.

3 Models

In this section, we explain the various neural network based models. First, we delineate our techniques and describe our models. We then analyze and discuss the results obtained.

3.1 Basic Bag of Words with Image Vector (BOWIMG)

For image-based features, we use a pre-trained model from the ImageNet 2014 contest winners (Simonyan and Zisserman, 2014). We use the VGG-ConvNet-19 by the Visual Geometry Group at the University of Oxford. The model was trained on the same images that are part of the VQA data set – making them appropriate for our usage. We use the activation outputs of the penultimate layer *(Dense layer)* of the *19 - layer* VGGNet as our features for the image with dimensionality of *4,096*. For all our experimentation, we keep this pre-trained model vectors fixed and they are not updated during training.

Each question has been converted to a Bag Of Words (BOW) and represented as a feature vector using Stanford's GloVe (Pennington et al., 2014) word representation vectors trained on 6B tokens from Wikipedia 2014 and Gigaword 5. The input feature vector for our MLP model is thus the concatenation of the summed word embeddings of the question and the image feature vector of size *4396* features.

We restricted the answer space to the top-1000 most frequent answers chosen from the training set. The problem reduces to multi-class classification and as the top-1000 most frequent answers represent 82% of the answers in the data set, we can expect the system to give a reasonable performance.

3.2 MLP Model

For the first part of our experimentation, we use a feed-forward multi-layer neural network with the input as the concatenated vector of the image representation and the sum of GloVe embeddings of the question. Thus, the input to our MLP formulation is of size *4,396* = 4,096 for the image feature and 300 for the question vector. Our model consists of two fully-connected Dense layers of size 1,000 units with *tanh* non-linear activation and *0.5* drop-out factor for each layer. This second layer is then fed to a fully-connected output layer of size *1,000* with a *softmax* activation method. Each cell in the output layer corresponds to the (Top) 1,000 answer vector distribution. Thus, by using *softmax* activation we get a probability distribution over these answers as the final output of our model. We use the "categorical cross entropy" as the loss function for our model and the "RMSProp" optimizer module provided by the Keras library. *Categorical cross entropy* minimization leads to the model giving best performance for the multi-class probability distribution.

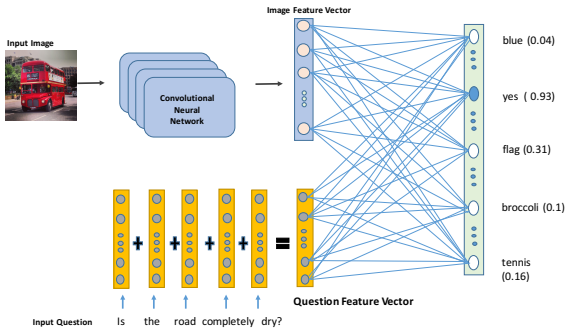

Figure 1: The architecture of the Learning model presented in the project

3.3 RNN Based Models

For the later part of our project, we look into recurrent neural networks with an objective that we see better understanding of the question structure. We experiment with LSTM (Long Short-Term Memory) to train a complex-model capable of understanding questions better than the linear sum of word embeddings that we pursued for the first part. LSTM was proposed by (Hochreiter and Schmidhuber, 1997) as an elegant solution of the vanishing/exploding gradients problems faced by researchers working on complex recurrent neural network. LSTM model introduces memory gates (forget gate, input gate and output gate) to regu-

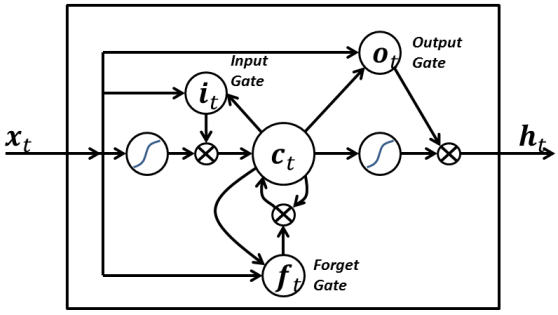

Figure 2: LSTM Unit. Image courtesy: WikiPedia

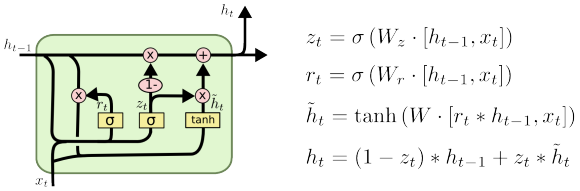

Figure 3: Gated Recurrent Unit. Image courtesy: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

late the weight/gradient update. Thus, LSTM are capable of supporting long-range dependencies in the input. Gated Recurrent Units are a (relatively new) variant of LSTM by (Cho et al., 2014). They have been gaining a lot of traction in RNN based models currently.

For our experiment, we use the LSTM/GRU based network to encode our question vector. We use two layers of LSTM/GRU cells of size 1,000 in a time sequence fashion. At each step, the input is the word embedding of a particular word (in order) of the question. The maximum question size is 23. Hence, our time-sequence has 23 time steps, each step representing a word. We pad shorter question with *0's* to get a standard input size.

The final layer of the LSTM/GRU model (which combines all time-steps into a single question vector) is then concatenated with the image features and fed to a fully-connected feed-forward network. This final structure is similar to part 1 and ends with a *softmax* regulated probability distribution over our common answer vector. This model also uses the "categorical cross entropy" loss objective function and *RMSProp* optimizer method.

4 Experiments

4.1 MLP Model

The training experiments were run for 200 epochs with the training data containing 250k image-question-answer examples. The testing was performed on a separate validation set of 120k examples. The time taken per epoch was *90* seconds.

4.2 RNN Based Models

The LSTM and GRU models were each trained for 100 epoch each over the entire training data containing 250k image-question-answer examples. For the GRU model, the time per epoch was *550* seconds and the LSTM model took *850* seconds per epoch.

4.3 Machine Setup

Our experiments were performed partly on a laptop with NVIDIA GeForce 740M GPU and partly on two AWS EC2 large GPU instances. We used the Keras library (`www.keras.io`) for modeling, learning and evaluating neural networks. Keras is a deep learning library based on Theano (Bastien et al., 2012). All neural network computations were performed over the GPU by using the NVIDIA CUDA Toolkit v6.5.

4.4 Further Analysis

This section of the report deals with the extra experiments we ran on the dataset based on the initial results that we obtained on running the MLP and LSTM based algorithms.

4.4.1 Heuristics

The accuracy of 41.44 % that we got for the multi-layer feed-forward neural network using a Bag-of-Words model for the question was obtained by predicting the distribution for each question over the 1,000 possible answers and taking the answer with the highest predicted probability. The model is agnostic of the 'multiple-choice' task formulation. We enhance on this by making use of the 'choices' posed along with the question. In an effort to find out how poorly we perform because of this, we tried a heuristic where we consider only the words in the 'multiple-choices' and then consider the word which is assigned the highest probability by the algorithm. The idea is to always return one of the given 'choices' and not return random/unrelated answers. For example answers like the following could be avoided:

```
Question: Is there a hot dog in
the image?
Answer: american
```

We get a marginal improvement in accuracy: from 41.44% to 46.69% upon using the heuristic

Model	Accuracy	On Curated Test Set (perturbed questions)
MLP + BOWIMG (Q+I)	41.44	20.93
MLP + BOWIMG (Q+I) + Heuristics	46.69	24.2
LSTM (Q+I)	38.40	20.6
GRU (Q+I)	33.86	18.23

Table 1: Performances of different models

approach so that we always return an answer from the given choices. We see clear improvements in the object-identification like – 'Who?', 'Which sport?', 'Which animal?'. This is most likely because the favorable outcomes of the model have multiple features with high probability – features like color, number etc – by using the given 'choices' explicitly, we can narrow down our response to a *better* answer.

4.4.2 Accuracy: Good indicator of learning?

The VQA challenge has a fixed training set and a possible similarity between images and questions in the training and test set. Thus, getting higher accuracies is a simple task of seeing what works (heuristics or algorithms) as we demonstrated in the previous subsection where we added a simple change to the algorithm to increase the accuracy by close to 4 %.

The question that we are looking at is, *Is the accuracy an accurate indicator of learning ?* In other words, if the model answers a question correctly can we assume that the model has, to some extent, learned the information that question asks for? We designed a *new data set* to find this out. **We consider the questions that we answer correctly and add multiple questions that query for the same information as the original question did**. In some cases, we just perturb the wording of the question to check if it breaks the algorithm. Naturally, the original question is considered to be answered correctly only if the algorithm answers *most of* the *new (potentially perturbed) questions* correctly else we can attribute the original question correctness to be a fluke or insignificant. The following image, original question and new questions are shown as an example.

```
Question: What sport are they
playing?
Answer: baseball

New questions:
1. Are they playing tennis?
2. Are they playing basketball?
3. Is baseball being played?
4. Are they not playing baseball?

Expected answers vs Predicted
answers
1. No, Yes
2. No, No
3. Yes, Yes
4. No, Yes
```

As we see here, only two of the four new questions are being answered correctly. The fourth question, specifically, is just the negation of the fourth question and the algorithm makes a mistake on this. We observe this trend on other images and questions as well. When presented with the negation of a correctly answered question, the algorithm fails. We populated the data set for *200* such new questions and found out the new accuracy. The accuracy for the questions that we were previously answering perfectly was found to be 50.05%. That is, **our overall accuracy is actually half of our original accuracy**.

We also notice that the algorithm is sensitive to the bias in the data set towards 'Yes'. It predicts a lot more 'Yes' than 'No' on the set of 'Yes/No' questions and this might be because of the skew in the data set.

Question Type	BOW	LSTM	GRU	Heuristics
what color	**35.22**	36.20	7.41	28.03
what kind	22.1	25.16	14.43	**34.67**
what are	18.50	20.83	13.85	**36.96**
what type of	21.5	26.82	13.77	**34.35**
is the	62.77	55.32	61.07	**63.82**
is this	64.77	54.04	61.26	**64.86**
how many	34.22	**36.55**	11.62	33.11
are	62.88	50.39	61.12	**68.98**
does this	67.28	61.87	**70.06**	68.82
does the	68.63	60.67	68.84	**69.94**
where is the	6.09	6.39	4.31	20.46
is there	**86.78**	70.40	77.07	77.35
why	6.84	6.18	11.24	**13.19**
which	21.28	21.20	11.64	**32.86**
do	63.16	57.81	62.57	**63.16**
what does	10.56	**15.58**	4.95	15.45
what time	12.55	10.96	3.77	**20.38**
who	3.25	2.63	0.92	**20.89**
what sport	78.02	80.15	76.30	**83.97**
what animal	52.97	55.23	36.59	**66.38**
what brand	19.61	24.77	13.59	**25.84**

Table 2: Comparison of performances on the multiple-choice task for various question types on MS COCO-VQA

4.4.3 How well do other models learn?

We consider one of the most recent models for visual question answering (Zhou et al., 2015) to find out how robust the model is to perturbations in the question. The authors have a demo website where the viewer can choose from a set of random images and input a question to get the top three answers. Here is an example of what we tried on the demo at `http://visualqa.csail.mit.edu/`:

```
Question1: Is there water in the
image?
Predictions:
surfer (score: 9.69)
yes (score: 9.42)
waves (score: 9.14)
```

```
Question2: Is water absent in the
image?
Predictions:
surfer (score: 9.84)
waves (score: 8.79)
surfers (score: 8.61)
```

We found that it was surprisingly easy to make small changes to the question to get completely wrong answers from the algorithm.

5 Results / Analysis

We were able to successfully reproduce the baseline results for Bag of words (BOW) and CNN given by the VQA team (Antol et al., 2015). Using the training and testing procedure described in the earlier section, we achieved an overall accuracy of **41.1%** on the validation set.

We performed a deeper analysis of the performance on the model across different question and answer types. We have given a sample of interesting images, questions and answers in Fig. 1.

We notice that the model performs better on the following question types: 'Is there' and 'What sport' (refer to Fig. 2). Also, we see better than

average on 'Does', 'Are', 'Do', 'Is the', 'Is this' and 'What animal' question types. This could be because we learn more from the images because of the significantly larger number of features based on the image (4096) as compared to those based on the question (300). We do very poorly on reasoning-based questions ('Why'), questions that depend on external information ('Who', 'What brand', 'Where'). We do not do that well on questions that are based solely on the image either ('How many', 'What type', ..).

As for the performance on different answer types, we do significantly better on 'Yes/No' type questions. The results for the other two answer types is not at all convincing.

Another thing that we notice is that our RNN based models do not perform as well as the feed-forward MLP model. We see that for some categories, there LSTM based model does better than the MLP model – especially object-identification – but for a few categories, the performance is very bad. We plan to spend more time here trying to understand the behavior observed.

6 Conclusions and Future Work

The project gave us an opportunity to explore the exciting problem of Visual Question Answering. In due course, we have not only evaluated various architectural models but also have identified that accuracy cannot be the sole metric to quantify understanding of visual and natural language features.

In fact, as we demonstrated, learning models are sensitive to perturbations in the questions as they fail to capture the correlation between informative words in the question and the answer and that between the image features and the answer. Actual reasoning and understanding of the question and the image is an avenue for future research.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. 2012. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078.

Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. 2015. Are you talking to a machine? dataset and methods for multilingual image question answering. *CoRR*, abs/1505.05612.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

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

Mengye Ren, Ryan Kiros, and Richard S. Zemel. 2015. Image question answering: A visual semantic embedding model and a new dataset. *CoRR*, abs/1505.02074.

Kevin J. Shih, Saurabh Singh, and Derek Hoiem. 2015. Where to look: Focus regions for visual question answering. *CoRR*, abs/1511.07394.

K. Simonyan and A. Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. 2015. Simple Baseline for Visual Question Answering. *ArXiv e-prints*, December.

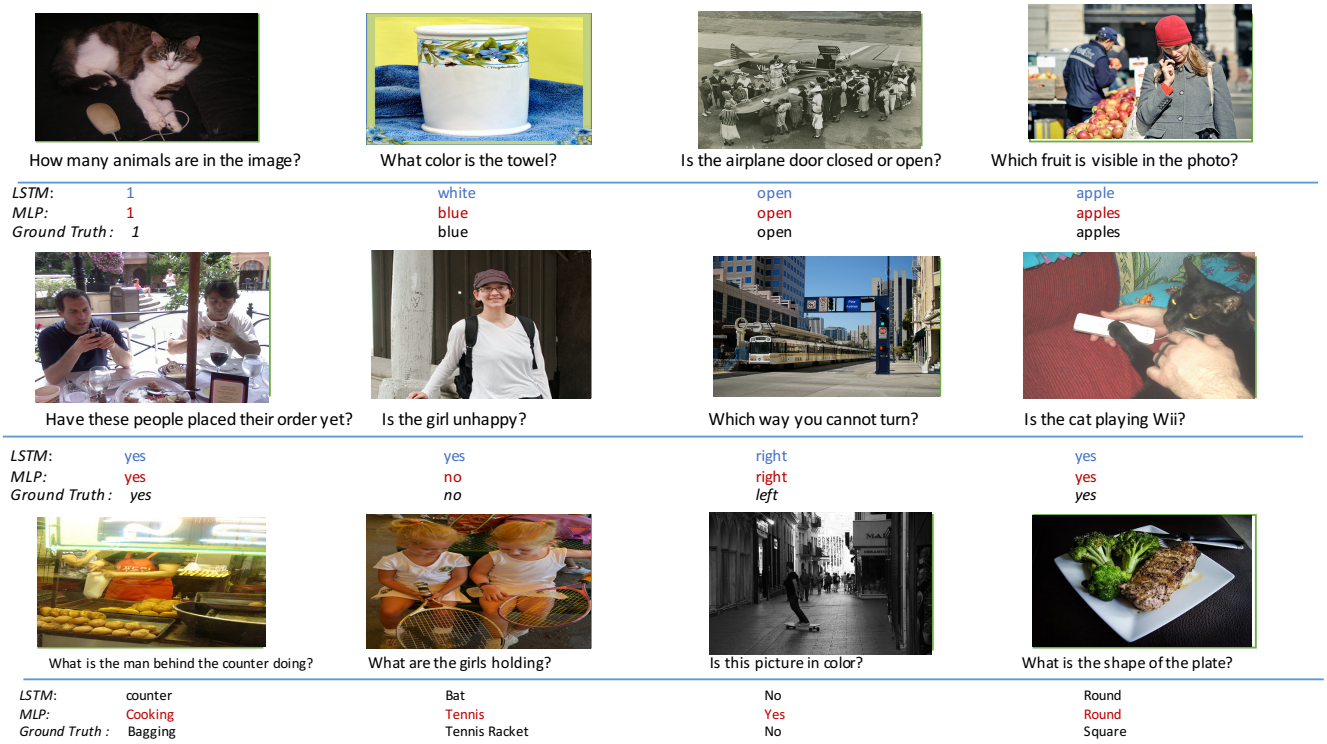

Figure 4: Comparison of examples betwen our final model and the VGGNet-LSTM. All results are from MS COCO-VQA.