AskAway: Visual Question Answering with Attention

Himanshu Aggarwal ha2467, Animesh Anant Sharma aas2325
Columbia University

***Abstract*—In this work we propose enlistment of DenseNet and LSTM/GRU based feature extraction models in a stacked attention architecture for Visual question answering. Given an image and a question, the proposed model was able to predict answers with comparable accuracy to the Resnet and LSTM based model proposed in [1]. The proposed model was able to achieve an accuracy of around 60% on the VQA 1.0 validation dataset [2] and around 59% on the VQA 2.0 validation dataset. The report also provides insights about the unbalanced nature of VQA 1.0 dataset and how the current state of the art methods are fixated on language priors in the sense that they do not even need the image input to provide an accuracy of around 49% on VQA 1.0 validation dataset for the same settings. The report further investigates the extent to which VQA 2.0 dataset [3] is balanced by repeating the same set of experiments under same settings. It was observed that even for the VQA 2.0 dataset, an accuracy of around 43% is obtained for validation dataset with image turned off and the execution time was reduced 2 times as compared to the normal settings. In light of these results, we hope that the unbalanced nature of the VQA datasets should be further remedied and the networks should be modified in such a way that such fixation on language priors be reduced.**

***Index Terms*—VQA, CNN, LSTM, Natural language processing, Attention, Deep learning Architectures.**

I. Introduction

Visual Question Answering (VQA) is a comparitively new problem in the field of computer vision and natural language processing. It focusses on inferring relevant and correct answers to text-based queries on a given image. Some examples of questions could be *"What is the color of batsman's jersey?"*, *"How many dogs are there in the scene?"* and then there can be more complex questions like *"What is there on the table along with the book?"*. For optimal performance, VQA requires a far deeper comprehension of scene semantics than other visual tasks like object recognition, object detection and object localization, tasks which need no understanding of the role of the object in the larger context. All existing VQA models encompass 1) Image featurization 2) Question featurization and 3) Algorithm to combine image and question features to generate answers. Image features are obtained from CNN models like VGG, Resnet or GoogLeNet trained on ImageNet dataset. Question features are extracted using LSTM encoders or simpler models like bag-of-words (BOW). Combining of image and question features can be done by simple concatenation, element-wise addition/multipulication or using more complex methods like encoding one form of features into a kernel which are then convolved with other features. Recently, Attention based VQA models have been proposed. Attention based models try to preserve task relevant regions by assigning few spatial regions in the image higher weights than other regions based upon question features, and thus creating a spatial attention in the image. One major issue with the datasets and proposed architectures for VQA task is that the methods tend to fixate on the language priors and this results in the visual feedback being not given much importance in final evaluation. This problem has been resolved to some extent in VQA 2.0 dataset [3] but this still remains an issue with VQA.

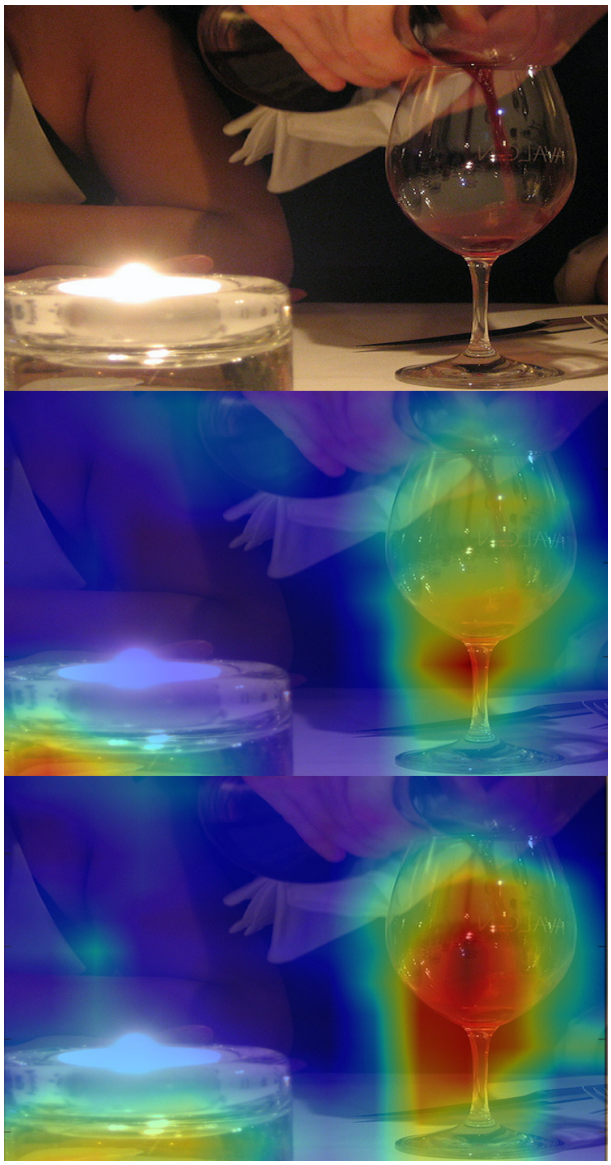


Fig. 1: Obtained attention from AskAway for question 'Does this flask have enough wine to fill the glass?' a) Input image b) First attention glimpse c) Second attention (finer) glimpse

II. Problem Formulation

In this project, we aim to make the following contributions:

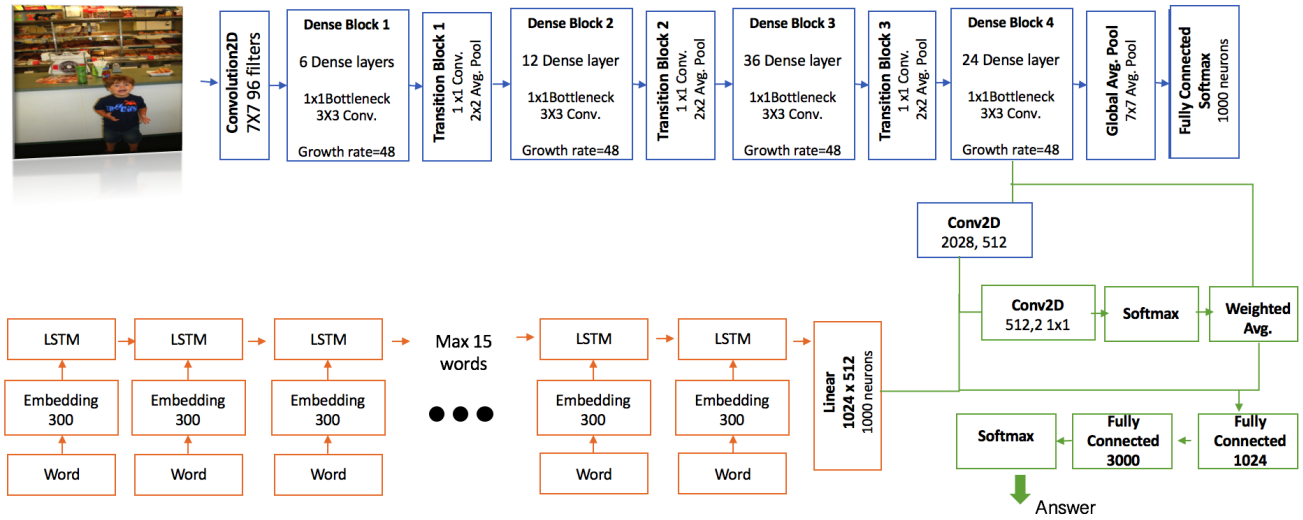


Fig. 2: AskAway VQA framework

1) Develop our own DenseNet based VQA system and compare it with Show, Ask, Attend, and Answer [1] Fig. 2.
2) Compare performance in above VQA sytems context, of two different Recurrent layers namely- a) LSTMs b) GRUs
3) Study the fixation of DenseNet and LSTM based system on language priors with the help of two experiments: (i) Replacing the image features provided by Densenet with a tensor of all 1s (ii) Removing the attention and image systems and transforming it into a simple question answering NLP task. This was first implemented for VQA 1.0 dataset [2].
4) Performing same language prior fixation experiments on the more balanced VQA 2.0 dataset [3].

III. Relevant Work

One task that is closely related to VQA is image captioning. Earlier papers [4] in VQA were based on modification of image captioning algorithms [5], which generated answer by conditioning LSTM [6] network on CNNs [7]. [8], used a bag-of-words to represent the question and CNN features from GoogLeNet [9] for the visual features. They then fed concatenation of these features into a multi-class logistic regression classifier. Similarly, [10] used skip-thought vectors [11] for question features and ResNet-152 extract image features and used MLP model with two hidden layers to generate answers. In [12], Bayesian framework for VQA was proposed. Semantic segmentation to identify the objects in an image and their positions, was used and a Bayesian algorithm was trained to model the spatial relationships of the objects, finally calculating each answers probability. Lately, fair amount of work is done on attention based models. The Focus Regions for VQA [13] used Edge Boxes [14] to generate bounding box region proposals for images. In [13], a CNN was used to extract features from each of these boxes. The input to their VQA system consisted of these CNN features, question features, and one of the multiple choice answers. In SAN [15], single layer of weights using the question and the CNN feature map with a softmax activation function was used to compute attention distribution across image locations. This distribution is then applied to the CNN feature map to emphasizes certain spatial regions of global features more than others.

IV. Proposed Method

We propose two different methods for VQA with the purpose of studying the fixation on language priors. The first one involves image pipeline and use Densenet to extract image features. Figure 2 illustrates the framework of this model. The second one experiments by flowing image of all 1s in the VQA pipeline. Alternatively, we also experiment removing the image and attention pipeline from the system.

A. With Image Pipeline Intact

Figure 2 illustrates the framework of our model. Given an image I and a question q in the form of natural language, we estimate the most likely answer a from a fixed set of answers based on the content of the image.

$$a* = \underset{a}{\operatorname{argmax}} P(a|I,q) \tag{1}$$

where $a = a_1, a_2, ..., a_M$.

1) Image Feature Extraction: We create visual understanding by extracting features from image using DenseNet architecture. To preserve spatial information, we use the output of the fourth Dense block of DenseNet-161 before global average pooling, rather than the final output. For an input image typically of size 448 x 448 x 3 (COCO images), the output feature vector is of size 14 x 14 x 2208. Table 1 shows all the calculated dimensions for each layer of DenseNet.
Same as in [1], we perform $L2$ normalization on the extracted features to enhance learning dynamics. The DenseNet has several advantages over Resnet which was used in [1]. It handles the vanishing gradient problem better and allows for feature reuse. So, it has about 3 times fewer parameters as

compared to Resnet for comparable accuracy. But the main reason to use DenseNet in context of VQA is that it was able to provide better accuracy as compared to Resnet on ImageNet challenge and thus supposedly provides better image features.

TABLE I: Feature dimension through layers of DenseNet

Layer	Spatial Dimension	Depth
Conv. Pool	112 x 112	96
Dense 1	112 x 112	384
Transition 1	56 x 56	192
Dense 2	56 x 56	768
Transition 2	28 x 28	384
Dense 3	28 x 28	2112
Transition 3	14 x 14	1056
Dense 4	14 x 14	2208

2) Question Encoding and Feature Extraction: To encode questions, we use embedding layer of 300 features for each word in the question, pass the output through $tanh$ non-linearity, before finally channeling it to Long-Short-term-memory (LSTM) recurrent layer. The final output of LSTM cell is of size 1024. To deal woth inputs of consistent size, we fix the length of input questions to 15 words. Questions longer than this are clipped while shorter questions are padded with zeroes.

3) Stacked Attention Stage: To generate attention over spatial regions of the image, we concatenate image and questions features. For this purpose, we alter depth of image feature vector to 512 by using 2D convolution. For question vector, we use a linear layer to convert 1024 dimensions to 512. Finally both the features are concatenated along depth. This concatenated vector is then fed to 2D convolution followed by softmax to generate 2 attention probabilities of spatiaal size 14 x 14. The original image features are then weighed using these generated proabilities to construct image glimpses. It should be noted that number of image glimpses generated is configurable and indicate granularity of attention as explained in the original stacked attention VQA [15]. Mathematically this can be represented as:

$$p_{c,l} \propto \exp\ G_c(q, \theta_{I,l}) \tag{2}$$

where $q = RNN(Q)$ for question Q and $\theta_I = DenseNet(I)$ for image I.

$$\sum_{l=1}^{L} p_{c,l} = 1 \tag{3}$$

$$W_c = \sum_{l} p_{c,l} * \theta_{I,l} \tag{4}$$

Each image feature glimpse W_c is the weighted average of image features θ_I over all the spatial locations $l = 1, 2, ..., L$. The attention weights $p_{c,l}$ are normalized separately for each glimpse c = 1, 2, ..., C. In practice G = $[G_1, G_2, ..., G_c]$ is modeled with two layers of convolution. Consequently G_cs share parameters in the first layer. We solely rely on different initializations to produce diverse attention distributions.

4) Classification and Answer generation: The generated image glimpses are concatenated with original question features and then fed to two fully connected layers of size 1024 and 3000 respectively. Finally, softmax is applied to generate probabilities over 3000 possible answers. The answer with highest score is then outputed. Again, the 3000 answer set is the set of most commonly occuring answers in the training dataset, and is a hyperparameter that can be tuned.

B. Without Image Pipeline

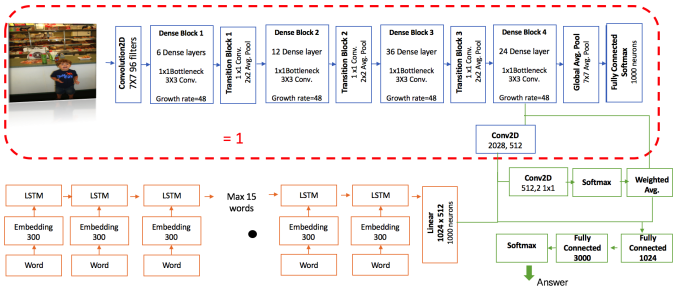


Fig. 3: Turning image vector to tensor of 1s to study fixation on language priors

1) Preserving model architecture: In the first part, we pass a tensor of all 1s as a image feature vector to evauluate the performance of the network when the image is turned off, as shown in Fig 3. No architectural changes are done in this study purposely, since it aims to be determine how the network behaves when it is visually blind.

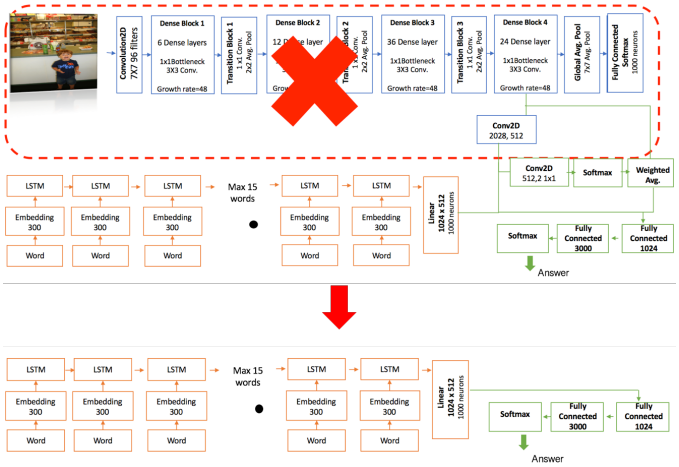
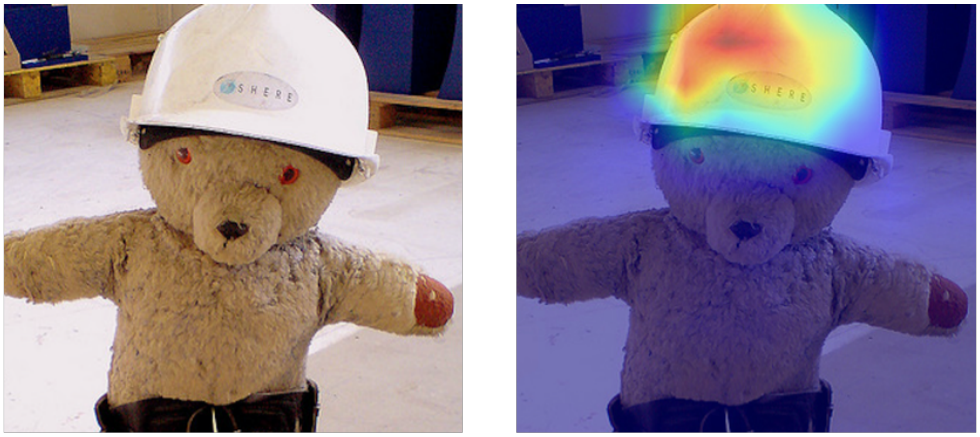


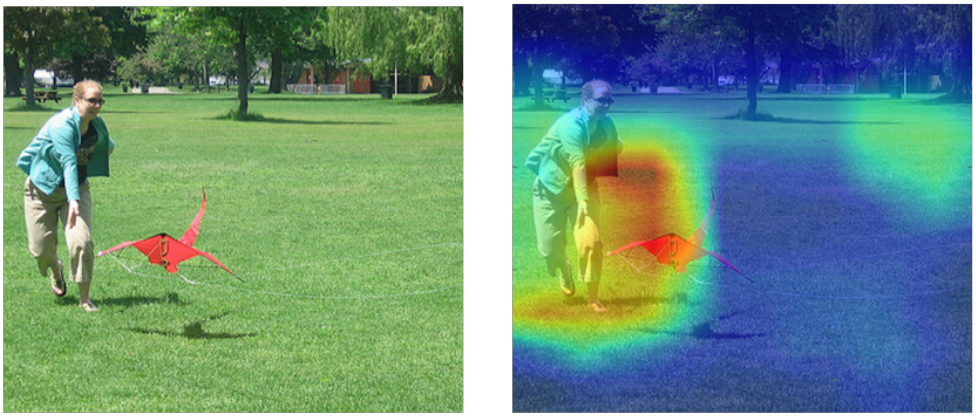
Fig. 4: Removing the image featurization architecture to study fixation on language priors

2) Changing model architecture: Alternatively, we completely remove the attention and image pipeline, transforming the problem into a Natural language processing task of question answering. The final model architecture is illustrated in Fig 4. The main idea here is to determine whether this simple network with reduced complexity and limited knowledge can provide the same accuracy. Also, the study should serve as a good measure of the fixation of VQA networks on language priors.



Question: What is the teddy bear wearing on the head? Answer: Cap

Question: What kind of meat is being prepared here? Answer: Hand



Question: What is the person doing to the kite? Answer: Throw

Question: What is written on the teddy bear's feet? Answer: Teddy

Fig. 5: Attention layers for various input cases. Red - Highest weights, Blue - Lowest weights

V. Evaluation and Analysis

A. Datasets

1) ***VQA-1.0:****:* This is one of the most widely used datasets in VQA. VQA 1.0 [2] contains 204,721 images from the MS COCO dataset. In contrast with the other datasets, binary (yes/no) questions were allowed here. There are 614,163 questions with 10 human annotators providing answers for each question. The dataset comes with predefined train, validation and test datasets. But this dataset is unbalanced in the sense that it presents bias which can result in a language only blind model to often guess correct answers with no visual input.

2) ***VQA-1.0:****:* This is a more balanced set as compared to VQA-1.0 and has 1,105,904 questions and 11,059,040 answers. VQA-2.0 [3] was recently released and this dataset has multiple images for the same question. This dataset has a one-to-many mapping from the question to the images such that the same question is repeated for multiple images. So the model cannot fixate much on the language prior here since otherwise the accuracy would decrease. The number of images remain the same in both the datasets.

B. Evaluation metric

For VQA datasets, we will use the following metric: min(# human labels that match answer/3,1), which means that full credit is given if the predicted output matches with response of atleast three human annotators. Otherwise partial credit is given. This can be calculated as:

$$Accuracy(a) = \frac{1}{M}\sum_{m=1}^{M} \min\left(\frac{\sum_{1\leq j\leq M, j\neq m} \mathbb{1}(a = a_j)}{3}, 1\right) \tag{5}$$

where $a_1, a_2, ..., a_M$ are the answers provided by the users. An average is being taken over 10 choose 9 subsets of ground truth answers.

The final loss calculated is similar to the softmax loss. It is given as:

$$L = \frac{1}{M}\sum_{m=1}^{M} -\log P(a_k|I, Q) \tag{6}$$

with $a_1, a_2, ..., a_M$ are the answers provided by the users.

C. Visualization of Attention Layers

To study the how well our model is performing, it is imperative to demonstrate the attention layers produced by our model for various test cases. Our model produces stacked attention, which leads to more fine-grained attention layer-by-layer in locating the regions that are relevant to the potential answers. In this work, we use two such levels of attention. To visualize the attention, we extract the $14 \times 14 \times 2$ attention probability tensor after the softmax layer in Fig 2. To better study the spatially attentive regions of input COCO images of size 448×448, we upscale the obtained tensor vector 32 times using bilinear interpolation.

The upscaled probability vector of size $448 \times 448 \times 2$ encodes layers of attention in the different channels (in this case 2). The different channels of the vector are then decoded into heatmap seperatel and superimposed on input images to generate visualizations. Fig 1. illustrates one such case of obtained visualization for the question 'Does this flask have enough wine to fill the glass?'. The red/ yellow parts of (b) and (c) show high attention regions. The color fades to blue as the attention weights reduce. It can be seen how in (b) the model first focusses on the candle along with the glass, because the candle looks like glass. However, in the second glimpse, the attention refines and put more weight on the actual glass. For this case, the model predicted answer 'No'. Fig 5 demonstrates more such visualizations.

D. Results

All experiments are done on Nvidia Tesla K80 GPU. Pytorch was used to train the network. Some of the common settings for the network are:

- We only consider top M = 3000 most frequent answers
- We use dropout of 0.5 on input features of all layers including the LSTM, convolutions, and fully connected layers.
- We optimize this model with Adam optimizer for 100K steps with batch size of 128. We use exponential decay to gradually decrease the learning rate
- The initial learning rate is set to α_0 = 0.001, and the decay steps is set to 50K. We set β_1 = 0.9 and β_2 = 0.999.

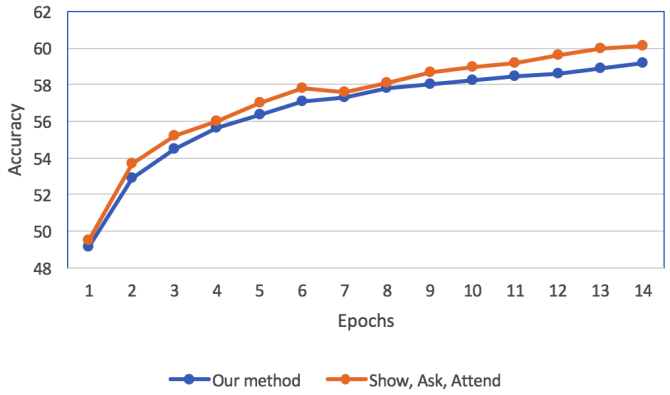


Fig. 6: Performance comparison of our method with [1]

1) DenseNet vs Resnet architecture: Due to high training time, we have completed only first 14 training epochs of for this experiment out of the originally planned 20 epochs. The results of the same for VQA 1.0 are summed up in Fig. 6. Table 2 lists the results and training time for this experiment. We have compared our obtained results with the current state-of-the-art, Show, ask, attend [1], which uses Resnet based feature extraction model. One other reason for stopping the training early was that the performance was almost same for both the architectures and as a results Densenet was used for further experiments. One reason we found behind the training time problem was the data transfer rate from hard disk to RAM being very slow as compared to the network computation speed. To remedy this, SSD was used for further

experiments and the number of data-workers were increased from 8 to 16. Table 2 shows results with SSD.

TABLE II: Results on VQA 1.0 for DenseNet and Resnet architectures for first 14 epochs

Params	Resnet[1]	Densenet	Densenet, GRU
Accuracy	60.1	59.0	59.1
Training time/ epoch	15 mins	15 mins	12 mins
Total training time	210 mins	210 mins	168 mins

We also experiment with GRU to obtain question encodings. The main difference with GRU is that it does not have an output gate and therefore has 2 gates [16]. The hidden layer is directly exposed. We found that convergence for GRU was a bit faster as compared to LSTM and the training time was less. Fig 7. illustrates the accuracy plot for this experiment on VQA 1.0

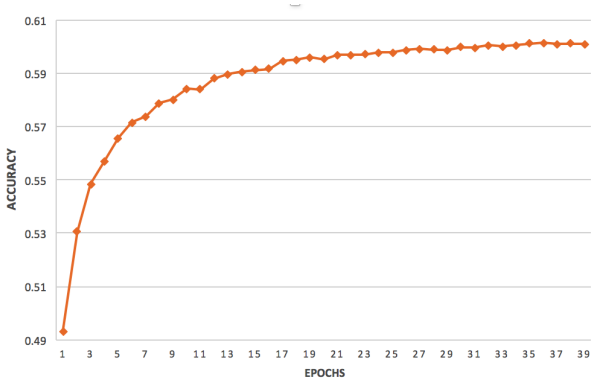


Fig. 7: Performance of GRU on VQA 1.0

We further trained our Densenet, LSTM model on VQA 2.0 to evaluate how much role image features play in accuracy. It was found that performance of our model did suffer on a more balanced dataset like VQA 2.0. Also, since VQA 2.0 contains more question/ answer pair, the training time increased considerably for this particular dataset. Fig 8 and Table 3 demonstrates the results.

TABLE III: Results on VQA 2.0 for AskAway: DenseNet, LSTM model

Accuracy	Training time/ epoch	Total training time
58.1	24 mins	480 mins

2) Without Image pipeline:

Image of 1s

The performance for both VQA 1.0 and VQA 2.0 is compared in Fig 9 and Table 4. We found, the training time to be almost same for the normal VQA experiment and this langauge prior experiment for both datasets. But the drop in accuracy for VQA-1.0 is from 60% to 49% while the drop for VQA-2.0 is from 59% to 43%. So it can be observed that VQA-2.0 is somewhat balanced but the fixation on language prior is there for both datasets.

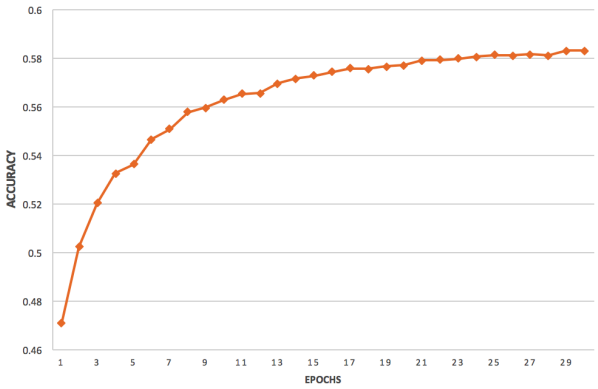
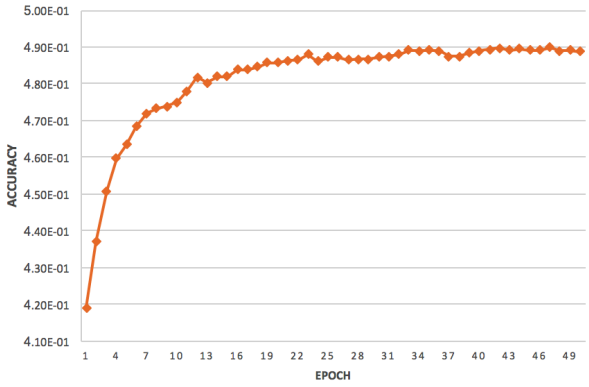


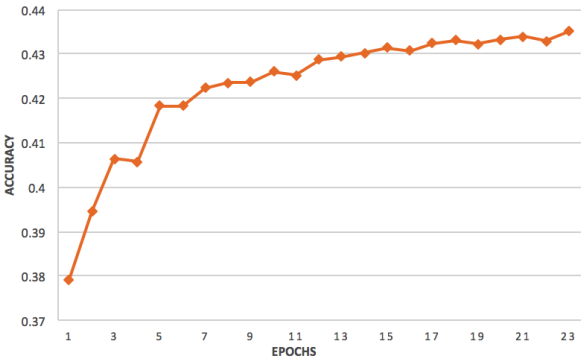
Fig. 8: Performance of AskAway: DenseNet, LSTM model on VQA 2.0

TABLE IV: Results of passing all 1s to Densenet, LSTM model on VQA 1.0 and VQA 2.0

Params	VQA 1.0	VQA 2.0
Accuracy	49.1	43.4
Training time/ epoch	15 mins	24 mins
Total training time	255 mins	360 mins



(a)



(b)

Fig. 9: Accuracy curve of passing all 1s to Densenet, LSTM model on a) VQA 1.0 and b) VQA 2.0

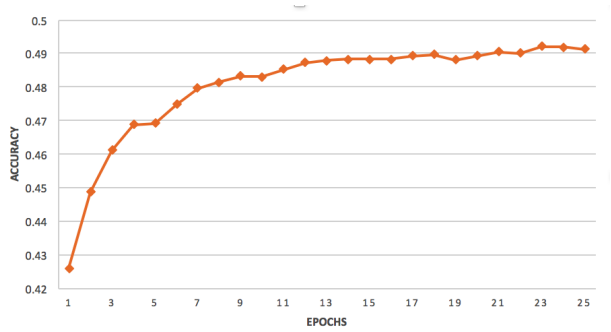
Removing image architecture

The performance for both VQA 1.0 and VQA 2.0 is compared in Fig 10 and Table 5. As expected, we found the results to

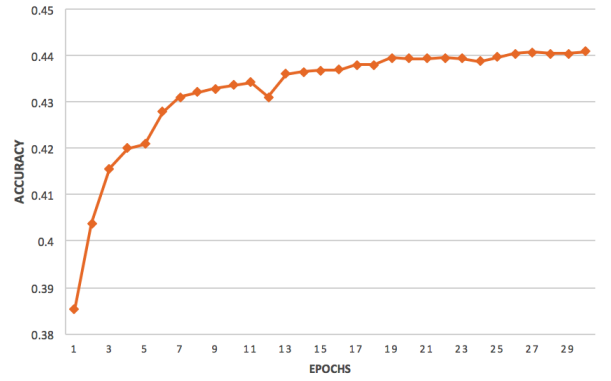
be very similar to earlier language prior fixation experiment. However, the main takeaway from this experiment is that the training time reduces by more than a factor of two, for both the datasets. This indicates that the simple NLP question answering setup can also provide considerably good results and this confirms the observation that the datasets are unbalanced for VQA tasks.

TABLE V: Results of removing image architecture in AskAway on VQA 1.0 and VQA 2.0

Params	**VQA 1.0**	**VQA 2.0**
Accuracy	49.3	44.0
Training time/ epoch	4 mins	8 mins
Total training time	45 mins	152 mins



(a)



(b)

Fig. 10: Accuracy curve of removing image architecture in AskAway on a) VQA 1.0 and b) VQA 2.0

VI. Conclusion

The proposed model was tested on both VQA-1.0 dataset and VQA-2.0 dataset and comparable results were obtained when compared with state of the art architectures. The GRU based implementation also gave comparable results with slight reduction in training time. The language prior experiments confirmed that the VQA datasets are unbalanced in the sense that the setup can predict the answers with really good accuracy without visual feedback. In future work, the datasets can be made more balanced or the methods can be improved in such a way that the visual information taken into account by the network have more influence on the final result obtained.

VII. References

1] Vahid Kazemi and Ali Elqursh, Show, Ask, Attend, and Answer: A Strong Baseline For Visual Question Answering, arXiv preprint arXiv:1704.03162, 2017.

2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In International Journal of Computer Vision, 2015.

3] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. CoRR, abs/1612.00837, 2016.

4] Ren, M., Kiros, R., Zemel, R.S.: Exploring models and data for image question answering. CoRR abs/1505.02074 (2015)

5] Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. arXiv preprint arXiv:1411.4555 (2014)

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

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

8] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, Simple baseline for visual question answering, arXiv preprint arXiv:1512.02167, 2015.

9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Van- houcke, and A. Rabinovich, Going deeper with convolutions, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

10] K. Kafle and C. Kanan, Answer-type prediction for visual question answering, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

11] R. Kiros, Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun, and S. Fidler, Skip-thought vectors, in Advances in Neural Information Processing Systems (NIPS), 2015.

12] M. Malinowski and M. Fritz, A multi-world approach to question answering about real- world scenes based on uncertain input, in Advances in Neural Information Processing Systems (NIPS), 2014.

13] K. J. Shih, S. Singh, and D. Hoiem, Where to look: Focus regions for visual question answering, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

14] C. L. Zitnick and P. Doll ar, Edge boxes: Locating object proposals from edges, in European Conference on Computer Vision, pp. 391405, Springer, 2014.

15] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola, Stacked attention networks for image question answering, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

16] R. Jozefowicz, W. Zaremba, I. Sutskever, "An empirical exploration of recurrent network architectures", Proc. 32nd Int. Conf. Mach. Learn. (ICML), pp. 2342-2350, 2015.