Image-Question-Linguistic Co-Attention for Visual Question Answering

Chenyue Meng
chenyue@stanford.edu

Yixin Wang
wyixin@stanford.edu

Shutong Zhang
zhangst@stanford.edu

Abstract

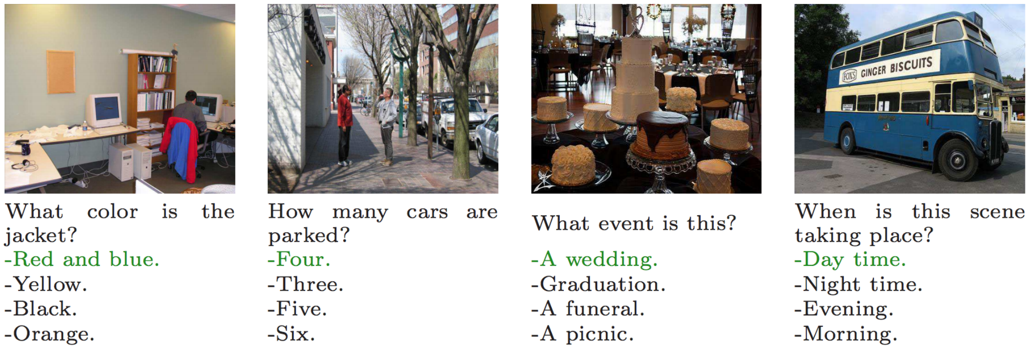
Our project focuses on *VQA: Visual Question Answering* [1], specifically, answering multiple choice questions about a given image. We start by building Multi-Layer Perceptron (MLP) model with question-grouped training and softmax loss. GloVe embedding and ResNet image features are used. We are able to achieve near state-of-the-art accuracy with this model. Then we add image-question co-attention [2] to exploit multi-modal relationship between pixels and words, which successfully boosts accuracy. Finally we add POS tag features on questions, and use an additional layer to capture the interaction between POS tag and word features, which further boosts performance. Experiment results on Visual7W dataset have shown the effectiveness of co-attention and POS tagging for VQA tasks. We also perform image and question co-attention visualization for further error analysis.

1 Introduction

With tremendous recent progress of computer vision, computers are now capable of dealing with complicated tasks such as object recognition, scene classification, action recognition[3]. However, the computer's ability of understanding *semantic* details of images still needs to be further evaluated on much complicated tasks. At the same time, the emerging of many successful deep structures in the field of natural language processing has lead to impressive performance in traditional Question-Answering problems[4].

Recently, academia and industry have been focusing on the problem of *VQA: Visual Question Answering* which is a intersection of computer vision and natural language processing. VQA is the task that for given text-based questions about an image, and the system needs to infer the answer for each question, where in our setting to pick an answer from multiple choices, shown in Fig 1. VQA has many potential real-world applications, among which one of the most valuable is to assist visually impaired individuals in understanding contents of images from the web. Moreover, VQA could be a great system to use in the Visual Turing Test [5] or other computer vision tasks to evaluate the system's ability when comparing with human performance.

Many challenges lie in this complicated problem. First, we have to extract embedding matrix for words in questions/answers and promising image features to support any further training on our proposed model. Also, whatever methods we use to extract multi-modal features, image features and text features are from different feature spaces which means it is very difficult to incorporate them in the same model. One possible solution is to explore correlation between multi-modal features or build relationships between these features. Bayesian models and visual attention models are most commonly used in state-of-art approaches.



What color is the jacket?
-Red and blue.
-Yellow.
-Black.
-Orange.

How many cars are parked?
-Four.
-Three.
-Five.
-Six.

What event is this?
-A wedding.
-Graduation.
-A funeral.
-A picnic.

When is this scene taking place?
-Day time.
-Night time.
-Evening.
-Morning.

Figure 1: Four images with associated questions and answers from the Visual7W dataset. Correct answers are highlighted in green.

But in our project, we think that it is not enough to only consider image attention, but it is essential to consider image attention based on question. Specifically, not only are we focusing on which region in the image we should look at, but which word and which POS tag feature in the sentence is of interest. In this project, we first implemented the baseline MLP model used in [6] and then by slightly changing the loss function we have the improved model. After that, we extended the co-attention model from [2] to be Image-Question-Linguistic co-attention and incorporated it into our model. Our experiments are conducted on a recently published real-world dataset Visual7W [7]. From the experiment results, when compared to baseline models, we can see significant improvement when using this advanced co-attention model to choose the correct answer from multiple choices.

The remainder of this paper is organized as follows. In Section 2, we provide a literature review for the field of *VQA: Visual Question Answering*, where the state-of-methods includes Bayesian models and visual attention models. In Section 3, we describe our Image-Question-Linguistic Co-attention Model in detail. The baselines and Visual7W dataset we use for this paper is analyzed in Section 4. Experimental results are also summarized in Section 4, where we can see the effectiveness of adapting co-attention model to the task of VQA compared of traditional structures. Finally, we conclude our work in Section 5 and provide ideas for future work.

2 Related Work

We will be exploring three main categories of related works on VQA and compare them with our proposed model.

The baseline algorithms of VQA are mostly combination of existing computer vision and NLP deep structures. In [8], CNN features from GoogLeNet and Bag-Of-Word representation of questions are extracted. Then they are concatenated together to form the input data. The features are fed into multi-class logistic regression classifier to generate the final answer. [6] is another example where Multilayer Perceptron (MLP) is used as classifier to choose the correct question-image-answer based on concatenated ResNet-101 features [9] and Bag-Of-Word features of both questions and answers. These baseline models have great performance but concatenating multi-model features does not consider the correlation of these features thus could certainly be improved.

In general, the problem of VQA requires drawing inferences and understanding relationships about image and question. Bayesian models have the ability of capturing the relations of image and question. Thus some works on extended Bayesian models have been published for the VQA task. Such model can be seen in [10]. The authors used semantic segmentation to determine the position of objects in image and computed the probability of different answers using a Bayesian model to capture relationships between question and image. The assumption of conditionally independence in Bayesian models made it difficult to further boost performance. Recent years, the capability of deep structures have been of interest for many other works on this problem.

Rather than using the entire image feature, many past works have taken visual attention into account to compute the ”weights” for different spatial image regions. Some also use text attention. The original idea behind attention model is that some parts of the image or some phrases in a certain sentence are more informative compared to other parts. A novel LSTM model with spatial attention is proposed in [7]. In this model, spatial attention terms are computed based on previous hidden states and the convolution feature map where only visual attention is considered. While on the other hand, [2] proposed to use co-attention of questions and images to locate the informative parts of an question-image pair. These co-attention terms could be trained inside a deep structure. Previous works have proven the effectiveness of attention models in the task of VQA.

Thus in our project, we mainly explore attention model on a recently published dataset. Compared to [7] and [2], in addition to question attention and visual attention, linguistic feature attention is also considered in our model. These attention terms are incorporated into the basic MLP structure proposed by [6] and are computed using some weight matrix during training process. Attention models could intuitively capture the relationships between questions and images, which will significantly help resolving the VQA problem.

3 Approach

3.1 Reviewing the Baseline MLP Model by Facebook AI

We start by implementing the MLP model in [6] because there is no currently available code for this baseline. This model takes image-question-answer triplet as input, and outputs the probability of this triplet being true. Both questions and answers are represented as Bag-Of-Word features of pre-trained word2vec embeddings. Images are represented as the penultimate layer feature of ResNet-101 [9]. These features are concatenated and fed into an MLP model with one hidden layer (8192 hidden units). Binary logistic loss is used. However, during experiments we found that binary logistic loss is very sensitive to the scale of scores. Sigmoid neurons saturate quickly when scores get large and thus did not train very well. Therefore we decided to slightly modify the model, described as follows.

3.2 Improved MLP Model

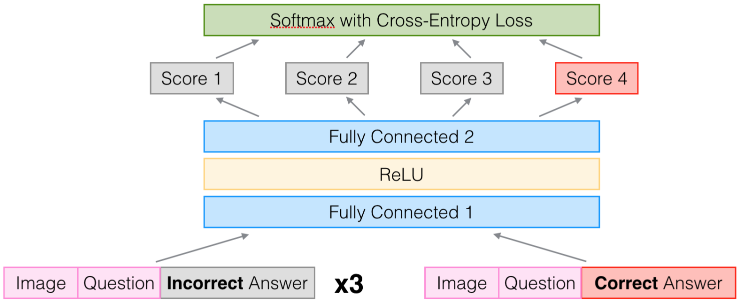


Figure 2: Illustration of our improved MLP model, trained as groups of 4, with 3 incorrect answers and 1 correct answer.

This model is based on Facebook’s baseline model but with several key differences, as follows. Firstly, questions and answers are represented by Bag-Of-Words feature of pre-trained GloVe embeddings [11], which are then fine-tuned in the training process. Secondly, we experimented two kinds of image representations: (1) The output of the penultimate (average pooling) layer of ResNet-101, and (2) Spatial average of the last convolution layer’s output. In both cases the dimension of image features is 2048×1. The most important difference is that, we do training in groups of four: one correct answer and three incorrect answers, and we then apply softmax with cross-entropy loss. Softmax is done over the output scores of all four answers. Compared with the binary logistic loss used by [6], softmax with cross-entropy loss is demonstrated to train better and converge more quickly in our experiments. A sketch of our model is illustrated in Figure 2.

Formally, we denote image, question and answer as $\boldsymbol{x}_i, \boldsymbol{x}_q, \boldsymbol{x}_a$ respectively, and the concatenated feature as $\boldsymbol{x}_{iqa} = \boldsymbol{x}_i \oplus \boldsymbol{x}_q \oplus \boldsymbol{x}_a$, where $\oplus$ is the concatenation operator. Each image-question-answer triple goes through the following MLP:

$$\boldsymbol{z}_1 = \mathbf{W}_1 \boldsymbol{x}_{iqa} + \boldsymbol{b}_1, \quad \boldsymbol{h}_1 = \max(0, \boldsymbol{z}_1)$$
$$\boldsymbol{s} = \mathbf{W}_2 \boldsymbol{h}_1 + \boldsymbol{b}_2$$

For each question, there are 3 incorrect answers and 1 correct answer, each of which is mapped to a score after going through the MLP. These four scores are then normalized using Softmax, and compared against ground-truth (boolean) labels to calculate cross-entropy loss.

3.3 Question-Image Co-Attention Terms

After implementing the improved MLP model, inspired by [2], question-image co-attention terms are being incorporated into our model to represent the relationships between image and question. Specifically, in the image we would decide “Where to look at” using the spatial attention terms, and as for the questions, we will have a sense of “What are we asking” in order to choose informative words within the sentence. The structure of co-attention model is shown in Figure 3 and we will describe it in detail as follows.

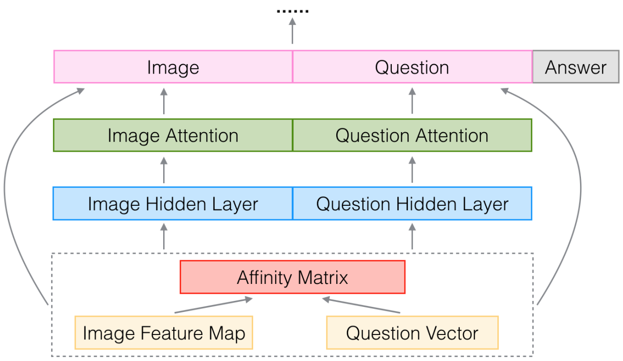


Figure 3: Illustration of adding weights computed by question-image co-attention

Denote the given raw image feature map $V \in R^{d_1 \times N}$ and question vector $Q \in R^{d_2 \times T}$ where N is the number of spatial regions and T is the length of the sentence. We will first compute the Affinity Matrix $C \in R^{N \times T}$:

$$C = tanh(Q^T W_c V)$$

where $W_c \in R^{d_2 * d_1}$. Then we will train the attention vector of image feature a_v and the attention vector of question feature a_q using an one-hidden layer neural network.

$$H_v = tanh(W_v V + (W_q Q)C), \quad H_q = tanh(W_q Q + (W_v V)C^T)$$
$$a_v = softmax(w_{hv}^T H_v), \quad a_q = softmax(w_{hq}^T H_q)$$

The dimensions for all the parameters mentioned above are $W_v \in R^{k \times d_1}$, $W_q \in R^{k \times d_2}$, $w_{hv}, w_{hq} \in R^k$. These terms are added on top of the raw feature layer to be used as weights. Then we could compute the weighted image feature and question feature as:

$$\hat{v} = \sum_{n=1}^{N} a_{v,n} V_n, \quad \hat{q} = \sum_{t=1}^{T} a_{q,t} Q_t$$

After that, the new weighted feature $\hat{v}$ and $\hat{q}$ are concatenated with the original answers discussed in 3.2 and will be fed into the structure for further training. During the training process, we found that the visual attention vector is not very smooth, thus we add a L2-regularization term on vector a_v.

3.4 Additional Hidden Layer with POS Tags Embedding

The intuition of incorporating POS tags is that, certain type of words in the question are worth paying more attention to. For example, intuitively nouns (NN), proper nouns (NNP) and adjectives (JJ) should be paid more attention to than determiners (DT) and pronouns (PRP). We adopt a structure similar to [12] in hope of capturing the interaction between words and their corresponding POS tags in each question. The structure is illustrated in Figure 4 and explained as follows.

For each word in the sentence $\boldsymbol{w}_1, \cdots, \boldsymbol{w}_n$, we use Stanford CoreNLP annotator [13] to annotate their POS tags $\boldsymbol{t}_1, \cdots, \boldsymbol{t}_n$. Then we represent each word as a d_w-dimensional vector $e_i^w \in \mathbb{R}^{d_1}, i \in [1, n]$ using pre-trained GloVe embeddings, and we represent each POS tag as a d_2-dimensional vector $e_j^t \in \mathbb{R}^{d_2}, j \in [1, n]$, trained from random initialization. In our experiments, $d_1 = 300, d_2 = 150$. These two vectors are concatenated as $e_i = e_i^w \oplus e_i^t, i \in [1, n]$, which is then fed into a fully-connected layer. We choose cube activation function [12] because it can model the intersection terms between input elements. (Note: Although it is sufficient to only model pair-wise intersection terms in our model, we did not choose square activation function because it is not monotone, which is not what activation functions should look like.) The activations are then averaged and used as a new feature $\boldsymbol{x}_{\text{pos}}$, which is concatenated with the image-question-answer features discussed in Section 3.2, and fed into MLP layers using the same structure as in Section 3.2.

Formally, the structure above is described as follows.

$$\boldsymbol{x}_{\text{pos}} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{W}_{\text{pos}}\boldsymbol{e}_i + \boldsymbol{b}_{\text{pos}})^3$$

$$\boldsymbol{x}' = \boldsymbol{x}_{iqa} \oplus \boldsymbol{x}_{\text{pos}}$$

In our experiments, we choose the dimensions of $\mathbf{W}_{\text{pos}} \in \mathbb{R}^{300\times 450}$. $\boldsymbol{x}'$ is the new feature which is then fed into MLP layers.

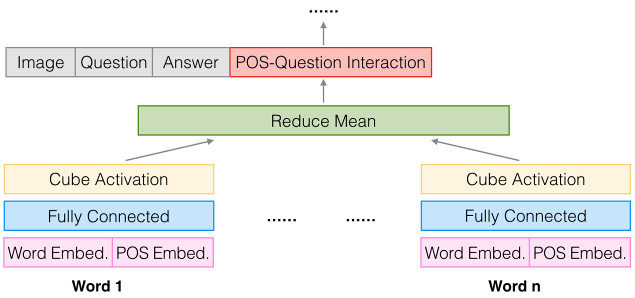


Figure 4: Illustration of how we model question and POS interaction. New features are concatenated with other features and fed into upper-layer neural network.

4 Experiments

4.1 Dataset

We evaluate our proposed model on **Visual7W Telling** dataset [7]. This dataset includes 69817 training questions, 28020 validation questions, and 42031 test questions. Each question starts with one of the six Ws, *what*, *where*, *when*, *who*, *why* and *how*, and has four answer choices. The three wrong answers are human-generated on a per-question basis. And the performance is measured by the percentage of correctly answered questions.

4.2 Experiment Setup

We implemented our model using Tensorflow [14]. Word embeddings are fine-tuned based on pre-trained GloVe [11] embeddings, and are of dimension 300. Image features are taken from forward propagation of ResNet-101[9]. We rescale the input images to 224 × 224 before feeding them to

ResNet. We use Adam optimizer with learning rate of 10^{-4}, which decays every 5 epochs with a base of 0.8. We set batch size to be 32. We apply dropout with keep-probability 0.5 after ReLU layer and use 10^{-6} as the regularization parameter.

4.3 Experiment Results

We evaluate the performance of the models described in Section 3, as well as their slight variants. These models are listed as follows.

(1) We implemented the improved MLP model (Section 3.2) and experimented with two types of image features. The first type is output of the penultimate (i.e. average pooling) layer of ResNet-101 (2048×1). The second type is taken from the activation of the last pooling layer ($2048 \times 7 \times 7$), and then averaged across spatial locations, reducing its dimensions to 2048×1. Experiments with these two types of image features are denoted as "Penultimate" and "Conv", respectively, in Table 1.

(2) We implemented question-image co-attention upon the improved MLP model (Section 3.3). Image feature is taken from the activation of the last pooling layer ($2048 \times 7 \times 7$), but multiplied with image attention, as is with question words. The corresponding experiment is denoted as "Conv-Att" in Table 1.

(3) We added new features to capture the interaction between question words and their POS tags. The new features are obtained with an extra hidden layer with cube activations (Section 3.4). Image feature is spatially averaging the activations of the last pooling layer. POS tag embeddings are vectors of size 150 and trained from scratch. The corresponding experiment is denoted as "Conv-POS" in Table 1.

(4) We stacked together the models in (2) and (3). The corresponding experiment is denoted as "Conv-Att-POS" in Table 1.

For each model above, we use the same parameters as specified in Section 4.2. The best model is chosen by taking the highest validation accuracy and evaluated on test set. We report test accuracies in Table 1. We plot the training and validation accuracy on "Penultimate" and "Conv-Att-POS" models in Figure 5.

Table 1: Overall accuracy and accuracy of each question type on Visual7W

Method	What	Where	When	Who	Why	How	Overall
LSTM-Att [7]	51.5	57.0	75.0	59.5	55.5	49.8	55.6
Penultimate	57.6	67.7	78.3	68.2	59.7	51.0	60.4
Conv	62.1	72.6	80.2	71.2	63.6	53.5	64.4
Conv-Att	63.1	**73.6**	80.4	71.2	64.4	54.0	65.2
Conv-POS	63.2	73.2	80.7	70.9	63.6	54.5	65.2
Conv-Att-POS	**63.7**	**73.6**	**80.8**	**71.3**	**64.5**	**54.6**	**65.6**

Additional Experiments

After comparing the test accuracies we got from the models we proposed with the most promising work [6], we found out that there is still an around 1.5% gap between our best results and theirs. As we mentioned before, the two main challenges we face for VQA problems are (1) the raw visual and text features need to be extracted (2) correlations between different feature spaces. However, using baseline structures similar to those from [6], the results are not as good. Thus, we decided to conduct additional experiments to analyze whether the input features may help us to improve the performance.

Table 4.3 shows the comparison of accuracy on our model with models proposed by [6]. To testify the effectiveness of the features, we fed only answer features and question/answer features into the basic MLP structure we implemented and compared the results with [6]. We can see from Table 4.3 that using only answer features and question/answer features, the performance of our model is even slightly better than MLP [6]. But after adding image features, our model's performance is not as good as MLP [6].

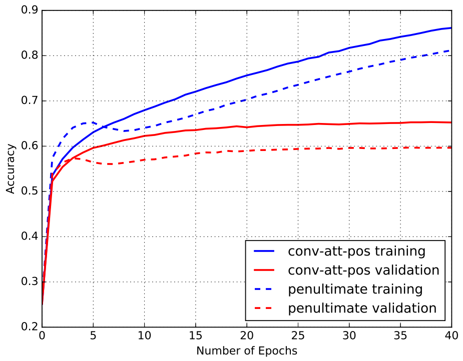


Figure 5: Training and validation accuracy of "Penultimate" and "Conv-Att-POS" model

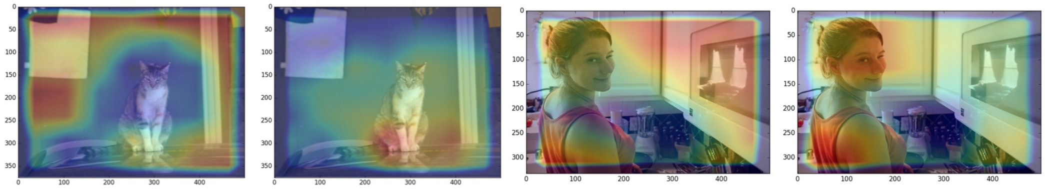
The reason behind this could be that we did not extract the exact same spatial features in ResNet101. Another possible cause could be our initialization of all the parameters are not the best choice. Or relatively large learning rate and less epochs caused by limited time since the authors in [6] trained the model for 300 epochs to gain the best performance.

Table 2: Comparison of accuracy on Visual7W of models from [6] (denote as MLP[6]) and our model "Conv-Att-Pos" based on different sets of input features

Features	**Models**	**What**	**Where**	**When**	**Who**	**Why**	**How**	**Overall**
A	MLP [6]	47.3	58.2	74.3	63.6	57.1	49.6	52.9
	MLP ours	48.4	58.4	75.2	63.3	55.9	50.7	53.6
A+Q	MLP [6]	54.9	60.0	76.8	66.0	64.5	54.9	58.5
	MLP ours	55.5	61.0	76.8	66.2	60.0	55.3	58.7
A+Q+I	MLP [6]	64.5	75.9	82.1	72.9	68.0	56.4	67.1
	Conv-Att-POS	63.7	73.6	80.8	71.3	64.5	54.6	65.6

4.4 Error Analysis

To better understand the advantages and shortcomings of our model, we performed visualization and error analysis on the best model we obtained in Section 4.3. All visualization and error analysis are done on the test set.



Q: What color is the wall directly behind the cat? **A:** gray

Q: What kind of animal is in the photo? **A:** a cat

Q: How many microwaves are there? **A:** one

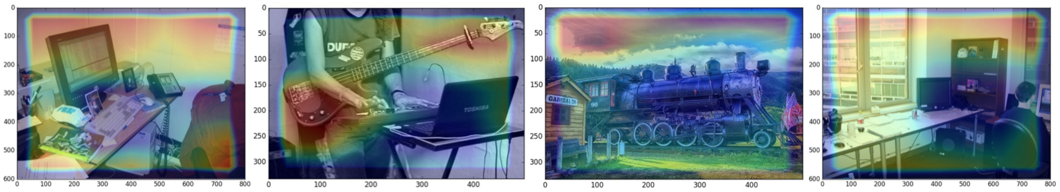
Q: Who is in the photo? **A:** a woman

Figure 6: Image and question co-attention. In each image, red region means higher attention and blue region means lower attention. In each question, we highlight one/two tokens with the highest attention.

Figure 6 visualizes image-question co-attention. Looking at the two image-question-answer triples on the left, we see that image attention is correctly placed in this example. Image attention is higher on the background in the first image because the question is asking about the background behind the cat, while image attention is higher on the cat in the second image because the question is asking about which animal it is. Corresponding question keywords also receive significantly higher attention than other words. Appropriate coupling of image-question attention helps the system make a correct choice. We can make similar analysis for the two examples on the right.

Then we analyze the errors that our system makes, as shown in Figure 7. In most error cases, at least one of image or question attention is incorrect. We found that the most common error in question attention is missing emphasis on certain important words. These errors occur when the question sentence is long or has complicated structures, e.g. when there are complicated prepositional phrases in the question, our system usually cannot handle it very well. The reason might be because using Bag-Of-Words or even weighted (with attention) Bag-Of-Words feature is not adequate for dealing with complicated sentences. Adding dependency parsing features might alleviate this situation. Using hidden layers features from RNN or LSTM might also help [15], because these models capture temporal information and thus can encode sentence structures better.

The most common error in image attention is emphasizing the incorrect region. This might be because ResNet is trained on the 1000 object classes on ImageNet, however there are more types of objects in Visual7W which ResNet is not trained on. Another problem is lack of granularity in image attention. This is because we were using 7×7 image spatial features instead of 14×14 as in [2], because of time constraints. However the second problem is not as serious as the first one.



Q: Who is in the picture leaning on the monitor? **WA**: an older couple

(a)

Q: What brand of laptop is being used? **WA**: mac

(b)

Q: What is in the background behind the train? **A**: trees

(c)

Q: What is on the wooden bookcase? **A**: a UNK

(d)

Figure 7: Errors in image and question co-attention. WA stands for the wrong answer our system produces. (a) Both image and question attention incorrect. (b) Correct question attention, incorrect image attention. (c) Correct image attention, incorrect question attention. (d) Problem of UNK token.

5 Conclusion

In this project, we focus on the task of Visual Question Answering where text-based questions are generated about an given image, and the goal is to pick a correct answer from four choices. We implemented a basic MLP structure as our baseline and further explored some tweaks to improve the model's performance. Then co-attention terms are computed as weights on question features, image features and POS tag embeddings and trained along with other parameters. Experiments conducted on real-world Visual7W dataset have shown the effectiveness of co-attention models, especially when we took image-question-linguistic co-attention into account. As for our future work, we can explore from these few aspects,

1. Because of limited resources and time, we were only able to use 32 as batch size and trained for at most 50 epochs for every set of parameters. In the future, increasing training time and batch size could possibly improve the performance of our proposed model.
2. During literal review, we found that several text attention models are proposed on top of LSTM. Right now, Our model only uses a very basic deep structure. Increasing the complexity of the proposed model may further boost the performance.
3. It may be helpful to find a way to incorporate intra-sentence attention and spatial attention into our model instead of only considering co-attention terms.

References

[1] K. Kafle and C. Kanan, "Visual question answering: Datasets, algorithms, and future challenges," *arXiv preprint arXiv:1610.01465*, 2016.

[2] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Advances In Neural Information Processing Systems*, pp. 289–297, 2016.

[3] R. Szeliski, *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[4] V. Lopez, V. Uren, M. Sabou, and E. Motta, "Is question answering fit for the semantic web?: a survey," *Semantic Web*, vol. 2, no. 2, pp. 125–155, 2011.

[5] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, pp. 3618–3623, 2015.

[6] A. Jabri, A. Joulin, and L. van der Maaten, "Revisiting visual question answering baselines," in *European Conference on Computer Vision*, pp. 727–739, Springer, 2016.

[7] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, "Visual7W: Grounded Question Answering in Images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[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] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[10] 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*, pp. 1682–1690, 2014.

[11] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation.," in *EMNLP*, vol. 14, pp. 1532–1543, 2014.

[12] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks.," in *EMNLP*, pp. 740–750, 2014.

[13] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit.," in *ACL (System Demonstrations)*, pp. 55–60, 2014.

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, 2016.

[15] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng, "Grounded compositional semantics for finding and describing images with sentences," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 207–218, 2014.