Baseline for Visual Question Answering

Jeroen Baars, Rik Helwegen & Sierk Kanis
10686320, 10516034, 10688528

Abstract

This work is meant as a validation of the results of [6], by comparing the baseline Bag of Words model with a basic Recurrent Neural Network. This was done by reaching 34,50% accuracy on a small subset of the COCO VQA data set with a BOW words model, and a 29.74% accuracy with a RNN. We conclude that, similar to Zhou et al. (2015), a Bag of Words model can reach comparable performance to a Recurrent Neural Network based approach.

1 Introduction

Artificial Intelligence (AI) has evolved to the point it can reach human-like accuracy, or beyond, on tasks in specific domains. A next challenge on the road of AI is combining domains [2]. Joining Computer Vision and Natural Language Programming seems promising, partly due to recent developments. In particular, Deep learning algorithms for Visual Recognition, such as [3], and availability of large corpses of Natural Language [4], enables researchers to create inter domain models. An application for this would be an autonomous agent which could communicate with humans in natural language about visual observations, a long-standing goal within AI [2].

In order to reach this, models for Visual Question Answering (VQA) are being developed [5], [1]. Zhou et al. (2015) benchmark a baseline model (iBOWIMG), feeding a concatenation of image and word features to a softmax layer, showing performance close to recently proposed Recurrent Neural Networks (RNN) [6]. Each question is fed to the model as a single vector representation, thus a bag of words (BOW) like approach. The improvement in cost and explainability of this simplification implies high potential for the field. Therefore, our work is meant to validate their results, by comparing their proposed baseline model with a basic RNN, and evaluate this by adding a more elaborate analysis of the two model predictions.

First of all, the taken approach for training the BOW and RNN is explained. Accordingly, their resulting performances are analyzed in both a quantitative and qualitative manner. The results are compared to the work of Zhou et al. (2015), and finally, a number of limitations are given together with a suggestion for future research.

2 Experiments

In this section the training of the BOW and RNN on a part of the COCO VQA data set is being stated. Like iBOWIMG [6], VQA is simplified to a classification task, with the space of predictable answers being the answer vocabulary of the training set.

In the COCO VQA dataset, there are 3 annotated questions per image in the COCO dataset. These questions have been annotated by humans to get 10 answers per question, then, by majority voting these answers, the most certain answer for each questions had been obtained. These answers, single or multiple words, were given as the target for our networks. The training set contained of 48061 pairs, the validation set 8977 pairs, and the test set 2962 pairs. The question vocabulary was reduced to size 5056, by stemming all words and removing unary occurring words, while the answer vocabulary was size 5176. The image features were computed using ResNet, and had a dimensionality of 2048. After hyper-parameter tuning, the train and validation set were combined to train the final models. The training of the models was terminated at the moment the minimum of the test-loss was reached.

2.1 Bag of Words

For the Bag of Words (BOW) the questions were encoded into one-hot vectors, with every word indicated by a one in the question vocabulary space, and the rest zeros. The Neural Network (NN) was matched to the network used for iBOWIMG, with an embedding layer that maps the question vocabulary space to a 164 dimensional embedding, and two linear layers that map the image features and the word embeddings to the answer vocabulary. The separate output of the image and the word part were summed up and activated by a SoftMax function. This separation was done in order to be able to separate the results based on the words and the images only, so the relative contribution to the total accuracy could be measured. The error of the target and the output was defined by the Cross Entropy Loss and was used to back propagate through the network. The final prediction was then made by taking an argmax over the output of the final output. The data was shuffled each training epoch and fed in the network in mini-batches of size 64.

Hyper-parameters to tune were the learning rate for the embedding layer, the learning rate for the Softmax layer, and the batch size. The search over the hyper-parameters was done by searching over batch-size [32, 64], lrembedding [0.01, 0.001, 0.0001] and lrsoftmax [0.001, 0.0001, 0.00001], with some combinations removed, since [6] states that the learning rate for the word embedding layer should be much higher than the learning rate of the softmax layer to learn a good word embedding. The final chosen hyper-parameters were [batch-size 64, lrembedding = 0.001, lrsoftmax = 0.0001].

2.2 Recurrent Neural Network

The implementation of the Recurrent Neural Network (RNN) was similar to the implementation of the BOW. The words of the questions were again encoded by a word-to-index dictionary, with the difference being that the words are forwarded through the RNN-LSTM layer separately, creating an hidden state per word. The output for the question, which is the hidden state of the last word, is then concatenated with the image features. This concatenation is put through a ReLu layer and finally mapped to a vector with the size of the vocabulary. The outputs were again separated, for the relative contribution to the accuracy to be measured. Notably, because of the non-linearity of the ReLu activation function, the separated outputs of the question and image features don't add up to the total output of the combined question and image features.

Again, hyper-parameters search was obtained in a similar way as for the BOW, with the final hyper-parameters being [batch-size 64, lrembedding = 0.01, lrsoftmax = 0.001, hidden-size = 128, embedding-size = 164].

2.3 Evaluation measures

The goal of VQA is having a model that can correctly answer a question about a given picture. Therefore, the accuracy of the predicted answer, compared to the humanly created annotation, is the usual evaluation for these kind of models. However, in order to create a better understanding of the performance, it is worthwhile to look further than only the highest ranked answer that the model returns. For example, a model which scores a high accuracy on having the correct answer in its top 5 predictions, could be considered more valuable than a model that has only a high accuracy for its best answer, but a low accuracy for all the sub-top answers. Ultimately, one could consider the distribution of the correct answer over the complete range of rankings, from now on called the *correct answer distribution*, see Figure 4 and 5. Utilizing this information, models can be compared on a more detailed level and this could reveal undesired behaviour.

Naturally, the distribution with most of its area being at the higher rankings are preferred. However, that does not translate to a distribution with the most first rankings, and therefore, the *correct answer distribution* was used as the final evaluation measure to chose the best model, instead of only considering the accuracy of the top-1 predictions.

3 Results

In this section the learning curves of the BOW and the RNN model will be shown, after which the models will be evaluated both quantitatively and qualitatively.

The models have made the following development while training, see Figure 1 and 2. Here one can see the difference in behaviour of the learning curve of the test loss and the accuracy of predicted answers. The model at the epoch with the lowest

test loss was chosen as the final model.

Figure 1: Training progress for the CBOW model.

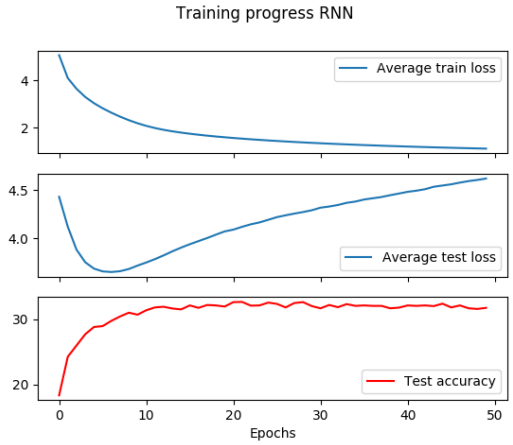

Figure 2: Training progress for the RNN model.

For both models, the training loss steadily descends, as is expected from a model which is able to learn.

Furthermore, it shows that for both models the test loss decreases to a minimum, after which it starts to rise again. A rising test loss during a decreasing train loss is known to indicate an overfit of the model.

Finally, in both cases the accuracy seems to raise at first, after which it stabilizes. The accuracy of the RNN model shows a more stable progression compared to the accuracy of the CBOW. It is worth to observe that the accuracy in both cases reaches its maximum at the moment the model starts overfitting, yet does not go down afterwards, even when its learned behaviour is overfitted on the training data.

3.1 Quantitative analysis

The optimal models resulting from above learning curves are quantitative analyzed in this section.

First of all, the obtained accuracies are given in overview in Figure 3. For both of the models, the accuracy is considered per question category. The accuracy of having the correct answer is shown, together with the accuracy of having the correct answer in the 5 highest ranking predictions of the model.

CBOW	Questiontype			
	Yes/No	Number	Other	Total
Accuracy %	19.72	7.49	7.29	34.50
Accuracy within question type %	57.25	22.98	22.13	34.50
Top 5 accuracy %	33.96	19.21	7.66	60.84
Top 5 accuracy within question type %	98.63	58.90	23.26	60.84

RNN	Questiontype			
	Yes/No	Number	Other	Total
Accuracy %	19.35	7.66	2.73	29.74
Accuracy within question type %	56.17	23.50	8.30	29.74
Top 5 accuracy %	34.23	19.72	3.14	57.09
Top 5 accuracy within question type %	99.41	60.46	9.53	57.09

Figure 3: Accuracy of BOW and RNN over different question categories.

From the tables in Figure 3, one can infer and compare the performance of the CBOW and RNN model. The CBOW model predicts the correct answer in 34.50 % of the test cases, whereas the RNN predicts the correct answer in 29.74% of the test cases. The CBOW and RNN have the correct answer in their top 5 predictions in 60.84% and 57.09% of the test cases respectively.

When the accuracies are compared per type of question, the two models show very similar behaviour for the 'Yes/No' questions and the 'Number' questions. For the 'Other' questions, the CBOW seems to outperform the RNN model. The performance difference for the 'Other' questions seems to be the main source for the performance difference in the 'Total' question

type between the models.

The above statistics limit the analysis of the models to only two points in the correct answer distribution, namely the top 1 and the top 5. To get a more elaborate insight in which rank of the predicted answers the correct answers are, the next figures provide the entire *correct answer distribution* for both the models, again per question type and for the total test set. For visibility, the plot shows only the first 40 ranks.

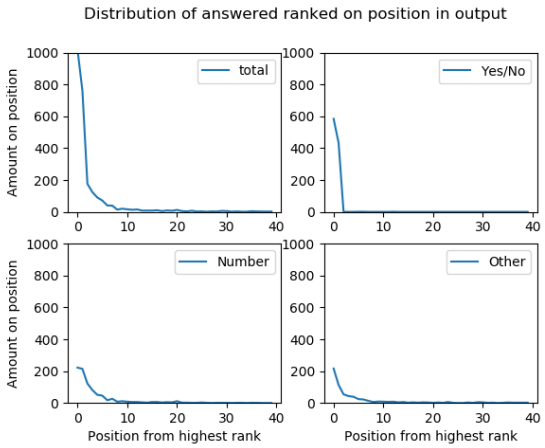

Figure 4: Correct answer distributions for CBOW

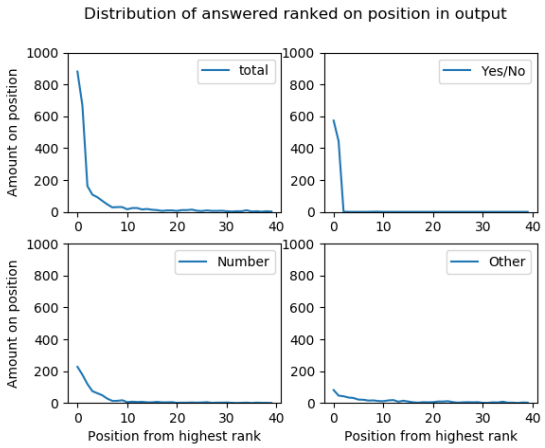

Figure 5: Correct answer distributions for RNN

As already noticeable in Figure 3, out of the different question types 'Other' questions shows the biggest difference between the models.

The additional value of having this overview over the rankings is to see how the ranking of the correct answer is distributed per type of question. For example, at the 'Yes/No' questions, one can see a clear structural break between the ranks 1, 2 and the rest of the ranks. Another observation to notice is that the 'Other' and 'Total' distribution of the RNN have a thicker tail compared to the 'Other' and 'Total' distribution of the CBOW model.

3.2 Qualitative analysis

To get a more intuitive understanding of how the models operate and differ, this section provides a qualitative analysis.

For both of the models, a number of example cases were picked out to show a particular behaviour of the models, see Figure 6. The final score of the prediction is split up in the influence of the image and the question separately. Also, the top three predictions based on image only and words only are shown. In this fashion, some interesting interactions between the image and questions can be found.

In the answers for the first picture, one can clearly see that, based on the image only, it cannot distinguish that the question is about a number, and therefore just predict the most common answers. However, the word parts clearly understand that this question is about a number, and therefore reduce the possible answer space to only numbers. Contrastingly, the second picture shows that, based on the words only, the BOW model can only guess objects that are most commonly used for tricks, whereas the image can tell that there is a skateboard in the picture. The two parts combined nicely illustrate the interaction between the image and word data: the word part restricts the answer space to objects, in order for the image part to decide which object it is. Finally, in the third picture it is interesting to see that, based on the words only, the BOWs answer would be 'right', only for the image to correct it and thereby predicting the correct answer 'left'.

In all the results the word part is used in order to massively scale down the answer space, whereas the image part is used to fine tune that answer space, which makes the image content decisive for its final answer.

4 Discussion

The overall performance of the BOW was 34.50%, compared to that of iBOWIMG's [6] 61.68%. The RNN should, according to the literature, be able

Question: **How many towels are hanging on the wall?** *Correct answer*: **3**

BOW predictions

0 (score: 0.30 = -6.00 [image] + 6.30 [word])
2 (score: 0.17 = -6.65 [image] + 6.82[word])
3 (score: 0.01 = -7.08 [image] + 7.09 [word])

Based on image only:
yes(-2.46), no (2.86), 0 (-6.00)
Based on word only : 3 (7.09), 2 (6.82), 4 (6.41)

RNN Predictions:

1 (score: 2.07 = -0.26 [image] + 3.05 [word])
2 (score: 1.90 = -0.50 [image] + 1.90[word])
3 (score: 0.84 = -1.21 [image] + 0.84 [word])

Based on image only:
bathroom(-0.00), yes (-0.11), 1 (-0.26)
Based on word only : 1 (3.05), 2 (2.75), 3 (2.36)

Question: **What is the person using to do tricks?** *Correct answer*: **Skateboard**

BOW predictions

skateboard (score: -2.53 = -5.80 [image] + 3.27 [word])
snowboard (score: -4.71 = -7.86 [image] + 3.15[word])
ski (score: -4.84 = -8.77 [image] + 3.93 [word])

Based on image only:
no (-2.37), yes (-2.44), skateboard (-5.80)
Based on word only: surf (4.48), ski (3.93), walk (3.79)

RNN Predictions:

yes (score: -0.32 = 0.26 [image] + -0.85 [word])
no (score: -0.53 = 0.17 [image] + -0.67 [word])
skateboard (score: -0.88 = -1.36 [image] + -1.66 [word])

Based on image only:
yes (0.26), no (-0.17), 1 (-0.49)
Based on word only:
white (0.11), blue (-0.08), green (-0.28)

Question: **Is the athlete left of right handed?** *Correct answer:* **Left**

BOW predictions:

left (score: -0.80 = -6.58 [image] + 5.78 [word])
right (score: -1.31 = -7.26 [image] + 5.94[word])
glove (score: -3.50 = -5.78 [image] + 2.29[word])

Based on image only: yes(-2.15), no(-2.36),
baseball (-4.42)
Based on word only: right (5.94), left (5.78), ski pole (2.52)

RNN predictions:

baseball (score: 0.71 = 0.15 [image] + -2.13 [word])
frisbee (score: -0.18 = -1.53 [image] + -0.67[word])
bat (score: -0.54 = -1.06 [image] + -3.08[word])

Based on image only: no (0.21), yes (0.21), baseball (0.15)
Based on word only: blue (-0.31), white (-0.44),
black (-0.62)

Figure 6: Pictures with according predictions of the CBOW and RNN model.

to surpass the performance of a baseline BOW model, but our RNN scored a performance slightly lower than the BOW, 29.74%. The performance of the BOW based on only words is 27.48, whereas Zhou et al. reached a corresponding performance of 53.68%. The image data only scored 19.65 %, while 30.53 % should be possible according to Zhou et al.

This research was bounded to a few limitations. First of all, the amount of data to train the models on could be increased. Due to heavy computational cost, only a subset of the COCO VQA dataset was used, as stated in the *Experiments* section. From the training results it shows that both the CBOW and RNN model suffer from overfitting after a number of epochs. Therefore, it is expected that increasing the amount of data will increase performance.

Next to a necessary reduction of data, the computational limits also caused a simplification of the hyper-parameter search. The behaviour of the models still varied between the different combinations of learning rates. Ideally, one would narrow down the intervals between learning rates in the search, in order to get closer to the optimal set of hyper-paramters.

To decrease the sparsity in the input, words which only occurred once were replaced by a paddy word. Treating this occurrence as a hyper-parameter might also increase performance.

For the final models, minimizing the test-loss was used as the evaluation method. The results in Figure 1 and 2 show that this is not equal to maximizing the accuracy. While being more overfitted, the CBOW and RNN reached a top 1 (correct answer) accuracy of over 37% and 32% respectively. Generally, this top 1 accuracy might be the main objective. However, as we see, the test-loss is already far from optimal when the accuracy converges, and so one can infer that this higher top 1 accuracy comes at the cost of a lower accuracy for lower rankings, the top 5 accuracy for example. When our model is used together with a Dialogue System (DS), we think our evaluation method of the lowest test-loss will prove to be critical. In general, models will likely never be able to be 100 % accurate, since it is sometimes not obvious what the correct answer would be. Contrastingly, a DS, with an input of an accurate top 5 accuracy, would then be able to

outperform any top-1-accuracy-evaluation based model. The combination of VQA and DS will be given as a suggestion for future research.

Concluding, the tested RNN and CBOW models have very similar performance. The RNN and CBOW perform similar for 'Yes/No', but the CBOW achieved better results for the 'Other' question category. These two question types mainly differ in sparsity of the set of answers. 'Yes/No' has the same answer occurring relatively often, whereas the 'Other' questions have a much broader range of answers. Therefore the conclusion can be drawn that the simple CBOW model is better at the handling of sparse data than the more complicated RNN model. This validates the research done by [6] in the sense that a baseline BOW model can reach comparable performance to several recently proposed recurrent neural network-based approaches.

References

[1] Haoyuan Gao et al. "Are you talking to a machine? dataset and methods for multilingual image question". In: *Advances in Neural Information Processing Systems*. 2015, pp. 2296–2304.

[2] Andrej Karpathy. "Connecting Images and Natural Language". PhD thesis. Stanford University, 2016.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[4] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[5] Mengye Ren, Ryan Kiros, and Richard Zemel. "Exploring models and data for image question answering". In: *Advances in neural information processing systems*. 2015, pp. 2953–2961.

[6] Bolei Zhou et al. "Simple baseline for visual question answering". In: *arXiv preprint arXiv:1512.02167* (2015).

5 Team responsibilities

The team for this research consisted of Jeroen Baars, Rik Helwegen and Sierk Kanis. Having different backgrounds, this was an excellent team to conduct this research. The individuals in the team succeeded to learn from each others strengths. The workload was equally divided over the group members.

Jeroen was expert on the obtaining, storing, structuring and querying of the data. Furthermore he made decisive contributions to the implementation of the LSTM network and the hyper-parameter search.

Rik conducted literature research to develop a theoretical background to work on. He contributed to the hyper-parameter search, implementation, the structuring of the results and to the discussion section.

Sierk kept the boat sailing the right direction. Always looking for the abstract and the details. He lay the foundations for both the models by starting small, made important theoretical decisions, implemented additional features in later stages, and managed to get priority on top in this paper.