# Ensemble based discriminative models for Visual Dialog Challenge 2018

Shubham Agarwal *†
Heriot Watt University
Edinburgh, UK
sa201@hw.ac.uk

Raghav Goyal *
raghavgoyal14@gmail.com

## Abstract

*This manuscript describes our approach for the Visual Dialog Challenge 2018. We use an ensemble of three discriminative models with different encoders and decoders for our final submission. Our best performing model on 'test-std' split achieves the NDCG score of* 55.46 *and the MRR value of* 63.77*, securing third position in the challenge.*

## 1. Introduction

Visual dialog [2] is an interesting new task combining the research efforts from Computer Vision, Natural Language Processing and Information Retrieval. While [6] presents some tips and tricks for VQA 2.0 Challenge, we follow their guidelines for the Visual Dialog challenge 2018. Our models use attention similar to [1] to get object level image representations from Faster R-CNN model [5]. We experiment with different encoder mechanisms to get representations of conversational history.

## 2. Models

Common to all the models, we initialize our embedding matrix with pre-trained Glove word vectors of 300 dimensions using 6B tokens [1]. Out of 11319 tokens present in the dataset, we found 188 tokens missing from the pre-trained Glove embeddings, so we manually map these tokens to words conveying semantically similar meaning, e.g. we map over ten variations of the word "yes" - misspelled or not picked up by tokenizer - "*yes", "yesa", "yess", "ytes", "yes-", "yes3", "yyes", "yees", etc.

For image features, we extract Faster R-CNN features with ResNet-101 backbone trained on Visual genome [3] dataset, similar to [1]. We use an adaptive number of object proposals per-image ranging from 10 to 100 generated using a fixed confidence threshold and each object is then associated with 2048-dimensional mean-pooled features using ROI pooling. We use discriminative decoding throughout our models.

We first describe our models individually and then the ensembling technique that we employ. In the following, *MN* denotes Memory Networks to encode conversational history, *RCNN* signify R-CNN for object level representations of an image, *Wt* represents additional linear layer in the decoder, and *LF* a late fusion mechanism as defined in [2].

**LF-RCNN** *Late fusion encoder [2] with concatenated history.* We use two-layered LSTMs with 512 hidden units for embedding questions and history. The object-level features are weighed using only question embeddings. The word embeddings from Glove vectors are frozen and are not fine-tuned. Figure 1 gives an overview of the architecture.

**MN-RCNN** *Memory network encoder [2] with bi-directional GRUs and word embeddings fine-tuned.* Object-level features are weighed by question and caption embeddings, and rest of the scheme is same as above. (See Figure 2)

**MN-RCNN-Wt** Same as above but with an additional linear layer applied to the dot product of candidate answer and encoder output, and gated using *tanh* function. Compare Figure 3 with Figure 2

**Ensembling** We ensembled final layer's log-softmax output - which is a distribution over candidate answers for each round (Figure 4). We use the three models described above and take the mean of the results (we also tried taking maximum of the results but found mean to perform better). We also tried ensembling a subset of the above three models, but found the combination of all three to outperform the rest.

* Both authors contributed equally for the challenge.
† Adeptmind Scholar
[1] https://nlp.stanford.edu/projects/glove/

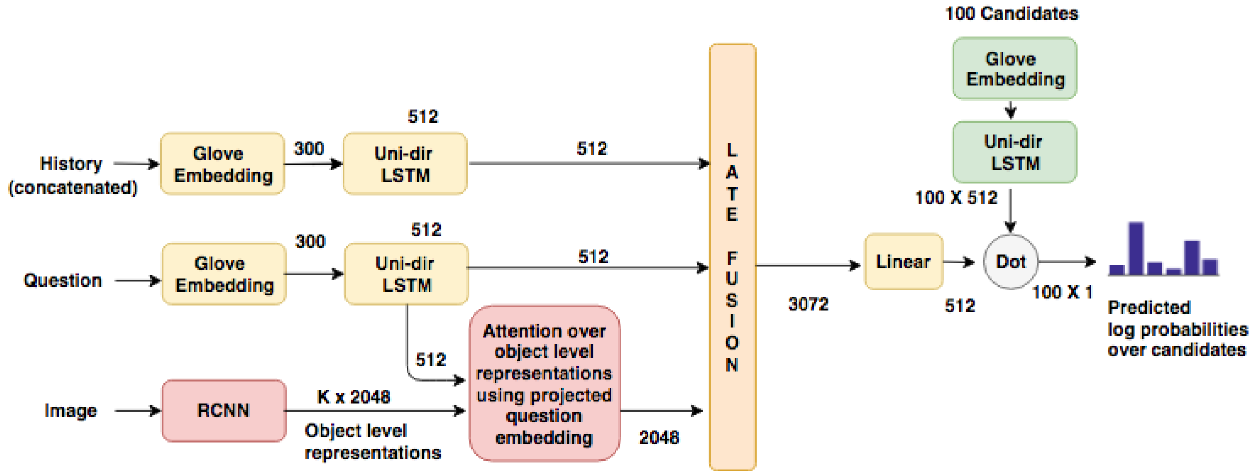


Figure 1. LF-RCNN model as defined in Section 2. Number indicates the output dimension of each layer.

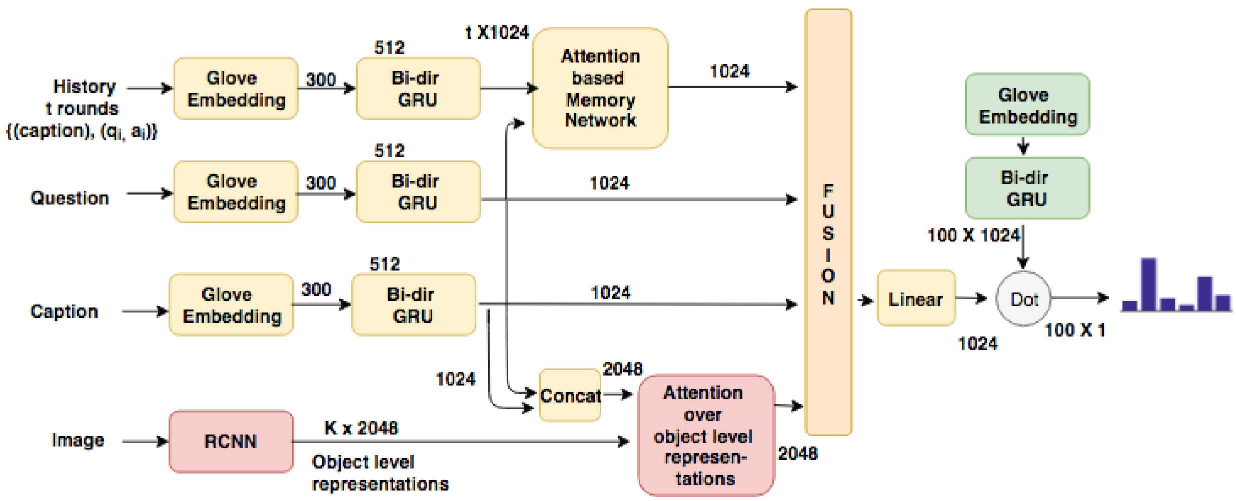


Figure 2. MN-RCNN model

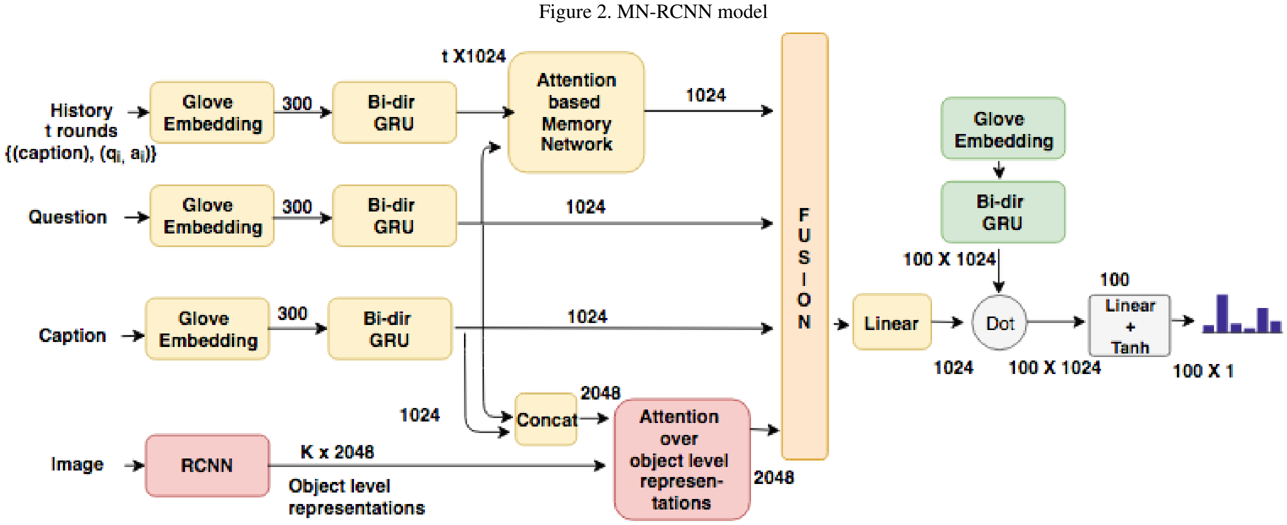


Figure 3. MN-RCNN-Wt model

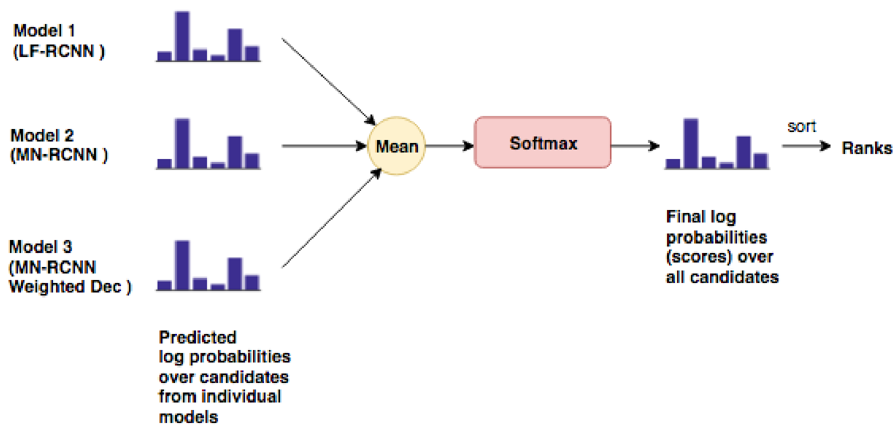


Figure 4. Ensemble using mean of log probabilities from the three individual models.

| Model | NDCG (x 100) | MRR (x 100) | R@1 | R@5 | R@10 | Mean |
|---|---|---|---|---|---|---|
| LF-RCNN | 51.69 | 61.03 | 47.03 | 77.83 | 87.55 | 4.70 |
| MN-RCNN | 53.59 | 61.25 | 46.78 | 79.43 | 87.93 | 4.63 |
| MN-RCNN-Wt | 53.20 | 61.50 | 47.10 | 78.7 | 88.38 | 4.54 |
| **Ensemble (all three)** | **55.46** | **63.77** | **49.8** | **81.22** | **90.03** | **4.11** |

Table 1. Results for the challenge on test-std. We took an ensemble of best performing models for the final submission.

| Model | MRR | R@1 | R@5 | R@10 | Mean |
|---|---|---|---|---|---|
| Baseline | 57.57 | 42.98 | 74.64 | 84.91 | 5.48 |
| MN | 59.24 | 44.64 | 76.48 | 86.41 | 5.14 |
| MN-Wt | 59.54 | 44.98 | 77.10 | 86.38 | 5.03 |
| LF-RCNN | 61.94 | 48.08 | 79.04 | 88.23 | 4.61 |
| MN-RCNN | 62.99 | 49.07 | 80.13 | 88.74 | 4.45 |
| MN-RCNN-Wt | 63.11 | 49.29 | 80.10 | 89.09 | 4.43 |

Table 2. Results on validation set. We show the impact of using additional gated linear layer in decoder. Compare MN with MN-Wt.

## 3. Experiments and Results

We used Pytorch[2] [4] for implementation.[3] In our experiments, we find that fine-tuning initialized Glove embeddings performed better than frozen embeddings. Object level representations play a critical role to generate a correct response from the model. Eventually, we use an ensemble of all the models described above for our final submission. Table 2 summarizes our results on validation set while Table 1 on *Test-Standard* split.

## 4. Conclusion

We experimented with discriminative models for our submission. Object level image representations gave a huge uplift in the evaluation metrics. Bi-directional GRUs constantly performed better than uni-directional LSTMs with Memory Networks outperforming Late fusion encoders for encoding conversational history. We even found that fine-tuning Glove embeddings performed better than their counterparts. Our final submission is an ensemble of three discriminative models and achieve the NDCG of 55.46 on test-std.

[2] `https://pytorch.org/`

[3] We are thankful to the organizers for providing a starter code in Pytorch on which we build our model. `https://github.com/batra-mlp-lab/visdial-challenge-starter-pytorch`

## References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, volume 3, page 6, 2018.

[2] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[3] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016.

[4] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[6] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge.