# ETH Zurich Visit Report

Visitor: Ted Zhang (KU Leuven)
Host: Dr. Dengxin Dai (ETHZ), Dr. Luc Van Gool (ETHZ)

Dates: Jan.23 - Feb.19, 2017

## 1 Description of Research

This brief report contains information pertaining to the progress I've made during a research visit to ETH Zurich. The work performed centers on visual question answering, which is the task of answering questions about images. To establish some reference points, I first implemented two baseline models.

### 1.1 Baselines

(1) uses image features from fully connected conv layers that consists of LSTM to encode question, CNN features to represent images, and merging of these two into a multimodal representation and is pushed through regular dense layers for classification. The architecture is inspired by the model from Lu et al [1] and is shown in figure 1.

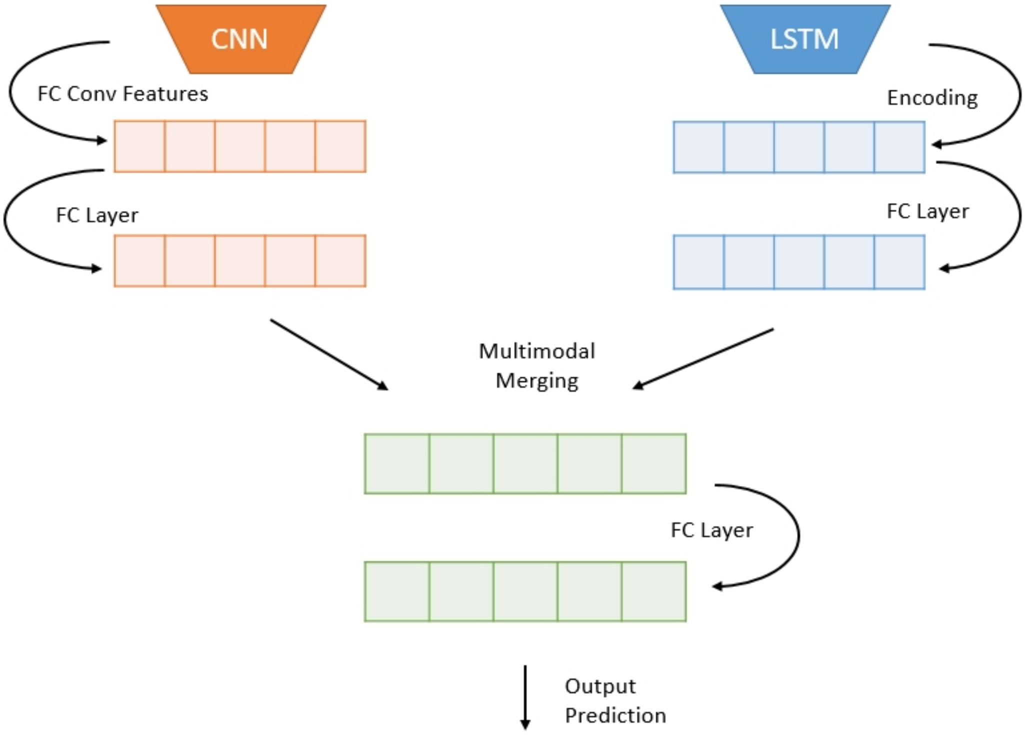


Figure 1: Model (1) using fully connected features. Performance achieved: 52.26% accuracy.

(2) uses image features from convolution layers: single layer of spatial attention that determines which spatial regions to pay attention to depending on the question encoding. The attention regions is weighted, then fused with the question encoding, and is pushed through regular dense layers for classification. Architecture is inspired by the stacked attention network from Yang et al [2] and is shown in figure 2.

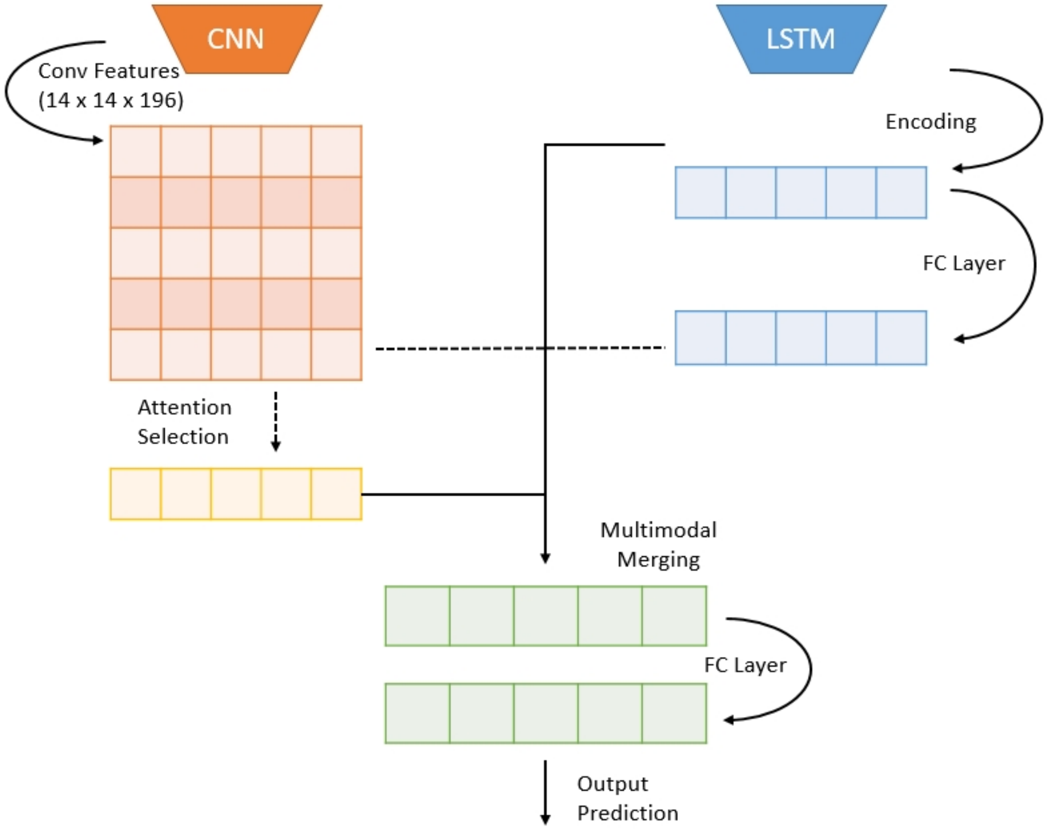


Figure 2: Baseline model (2) using conv features and attention mechanism. Performance achieved: 51.59% accuracy

## 1.2 Improvement in Architecture

I propose to improve the model by building an attention architecture that integrates multiple models, and pays more attention to the model that will likely output the correct answer for the given type of question. The model is shown in figure 3. For example, if model A performs well on 'what color is ...' type of questions, and model B performs well on 'where ...' type of questions, we should think that combining the strengths of these models will lead to higher outcomes than each of the individual model's performances. The challenge is how to know when one model would perform better than another, and how can one quantify how much weight one should put on one model as opposed to another?

I propose a soft gated mechanism that will take the LSTM encoding of the question, and map it to two variables, which add to 1. In other words, we have $x_1$, $x_2$, where $x_1 + x_2 = 1$, and $x_1$, $x_2 >= 0$. Staying with the example of models A and B, each model outputs their prediction as a vectorized softmax distribution of 1000 classes, which I call $p_A$, and $p_B$. I take each of these 1000 classes, scale them by $x_1$, and $x_2$, and add them. Concretely, the combined distribution of these two models after scaling are $(p_A{*}x_1) + (p_B{*}x_2)$. The final output of this integrated model is obtained after making another softmax such that the distribution sums to 1. One can see that if $x_1 = 1$, then $x_2$ must be 0, which means the model does not care at all about the predictions made by $p_B$. The same goes vice versa. This way, I have a way of automatic model selection without having to rely on manual parsing of the question. The neural net learns associating the strengths of each model with a type of question. For brevity, I refer to this model as a "merging model".

Before training this model, I initialize the weights of the two branches with the weights from the models when trained individually. At the time of writing, this method has been tested with 52.36% accuracy, which is an insignificant gain over the individual models. The next step is to try different types of 'deciding' gate features, instead of a binary weighting scheme.

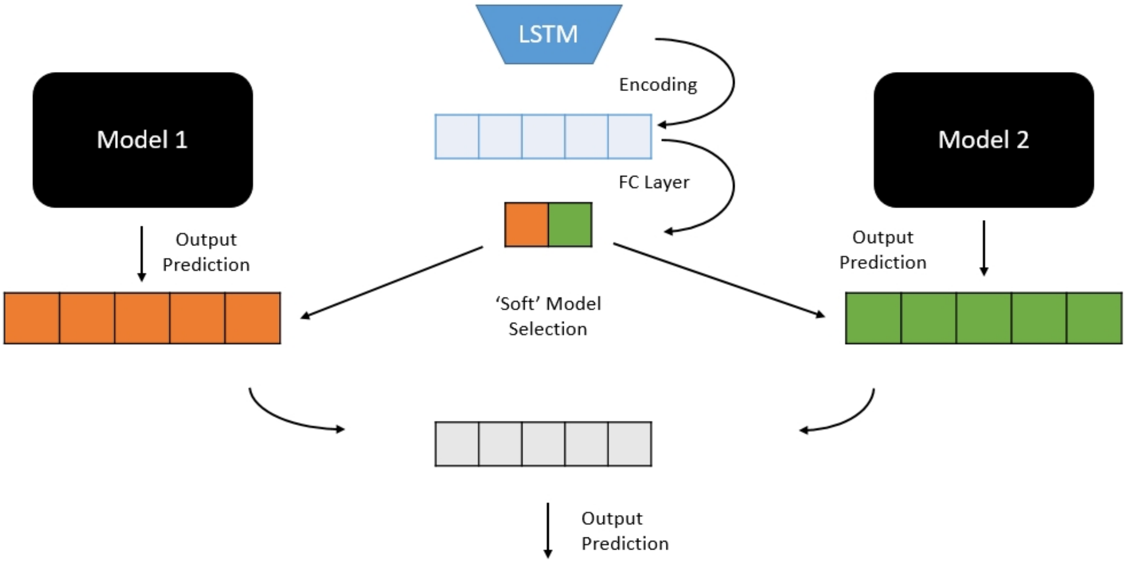


Figure 3: Baseline model (2) using conv features and attention mechanism. Performance achieved: 51.59% accuracy

## 1.3 Improvement in Data

Another direction to improve performance is to obtain more training data, via paraphrasing. For each image answer pair, I take the original question and generate another question similar to it that has the same answer. This way, I can cover a more variety of ways the question can be asked and have the model still be able to answer with the correct answer. Paraphrasing itself is a novel task, as it is still unsolved in the NLP community. I propose an unsupervised training method to teach a network how to paraphrase. This can be done via sequence to sequence learning.

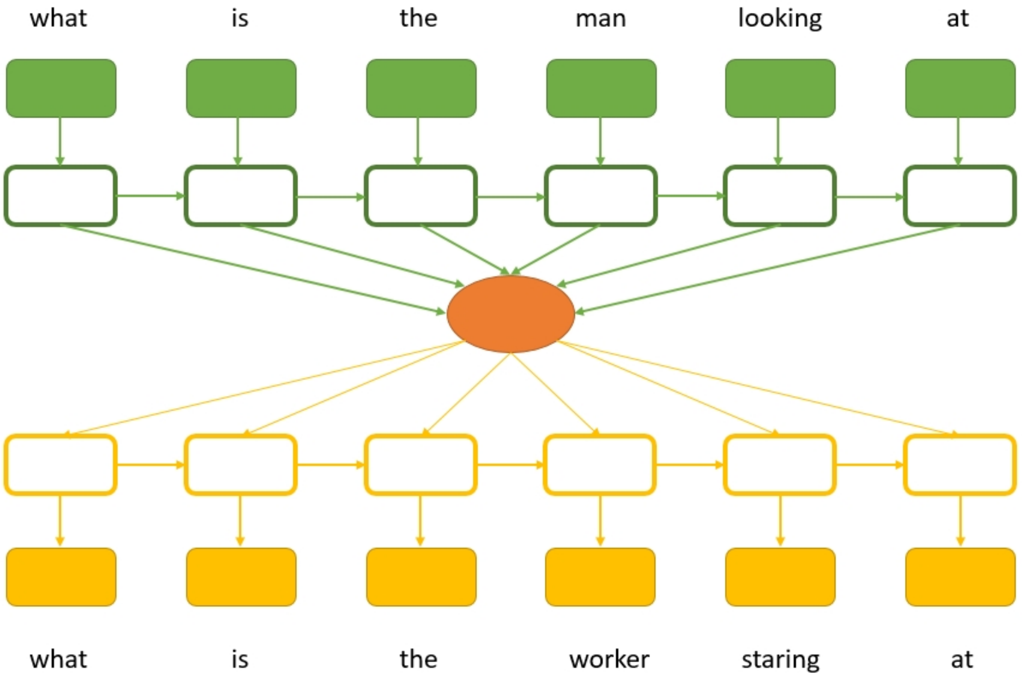


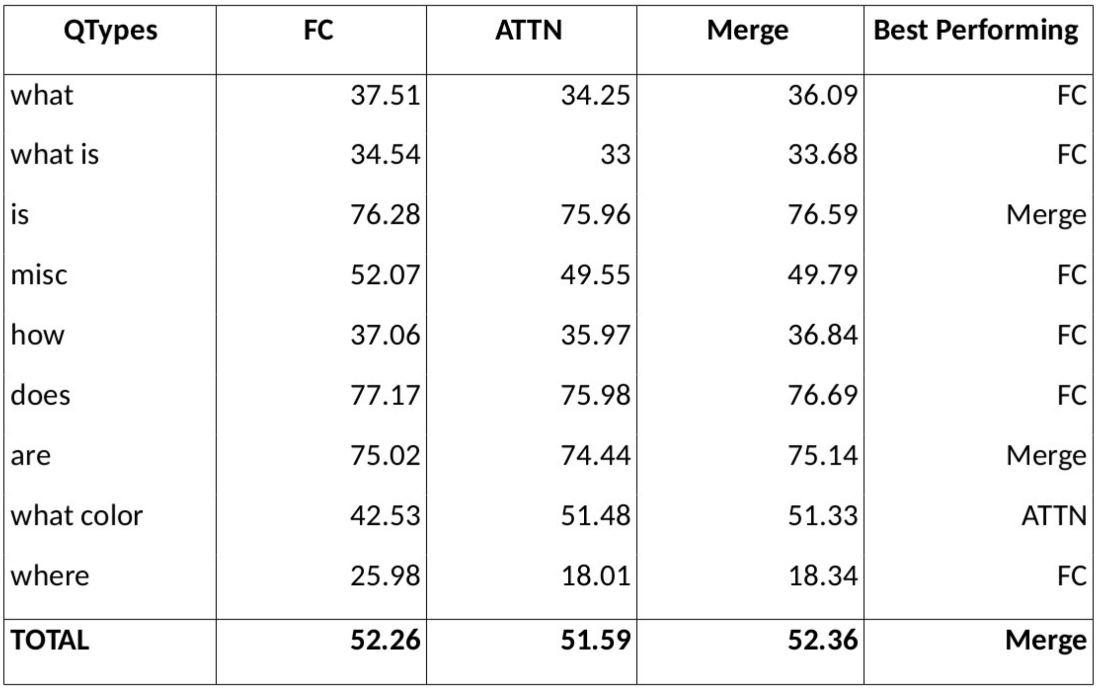
Figure 4: Unsupervised sequence to sequence using the same input as output. During prediction time, the output may be slightly different.

Sequence to sequence was first used for machine translation. A source language is encoded into a representation, and a decoder to takes that representation and generates the corresponding sentence in a target language. My proposed method takes as “source” some sentence, and tries to predict the exact same sentence as output, or “target”. In other words, the input and output are the exact same string. The model compresses the source into a compact representation, and then attempts to reconstruct the original string. This is in a similar spirit to autoencoders for denoising images, as one can see in the figure 4.

The idea here is that during reconstruction, the model may pick words that are semantically similar to those of the original words (since they are assumed to be neighbors in vector embedding space). When the model predicts the exact same output strings during training, this is good, and shows it is learning, but it is not so useful during test time. During test time, I want a paraphrase of the original sentence, so to do this I simply sample the output distribution. For example, the question “What is this animal?” may be sampled to output “What is this creature?” When sampled, “animal” becomes “creature”, because these two share similar values in latent space, and substituting these two gives a similar sentiment as the original sentence. However, sampling does not change “What is the” to for example, “Who is the”, because that is a distinctly separate sentiment and can be expected to be represented differently in latent space. At the moment, this system is being trained and can paraphrase but not with consistency. Once (or if) the system reaches a state of reliable paraphrasing, I will use this system to generate extra training examples and compare the performance of the model using the augmented dataset.

## 2 Main Results

Figure 5 shows the results of the different models. Note that using data augmentation to test performance has not yet been done.



| QTypes | FC | ATTN | Merge | Best Performing |
|---|---:|---:|---:|---:|
| what | 37.51 | 34.25 | 36.09 | FC |
| what is | 34.54 | 33 | 33.68 | FC |
| is | 76.28 | 75.96 | 76.59 | Merge |
| misc | 52.07 | 49.55 | 49.79 | FC |
| how | 37.06 | 35.97 | 36.84 | FC |
| does | 77.17 | 75.98 | 76.69 | FC |
| are | 75.02 | 74.44 | 75.14 | Merge |
| what color | 42.53 | 51.48 | 51.33 | ATTN |
| where | 25.98 | 18.01 | 18.34 | FC |
| **TOTAL** | **52.26** | **51.59** | **52.36** | **Merge** |

Figure 5: Comparing the performances between different models.

The model that uses fully connected features obtains an accuracy of 52.26%, while the attention model using convolution features obtains 51.59%. This is surprising, due to the fact that convolution features are much more fine grained and detailed in comparison with the fully connected features. However, this can be attributed to parameter tuning, as attention models are notorious for being difficult to train.

The merge model obtains 52.36%, which is an insignificant performance over the two models it is composed of. This figure is premature, as more work needs to be done on the different merging strategies and initialization methods, i.e. whether to freeze the layers of the models or not.

Lastly, the analysis of what types of questions one model gets right vs another is quite surprising, especially in the where category. One would expect the attention model to localize better, since it retains geometric information in its convolution features. However, the FC baseline model performs 8% better in this category.

# 3 Future Collaborations

## 3.1 Concretize Results

The results in the previous section are as mentioned before, still preliminary, and thus I plan to work with ETHZ to finalize the results on the test sets. Furthermore, the method of data augmentation is still in its prototyping phase. I will maintain communication with ETH to concretize its implementation to ensure the paraphrases are realistic.

## 3.2 Speech and Vision

Another idea that I'm currently working on with ETHZ is question answering using speech instead of text. That is, instead of the question being in the form of a sentence of text, it is in the form of an audio snippet. The idea here is to explore whether an end to end speech answering system is feasible, instead of having to convert speech to text first.

# 4 Planned Publications

At the moment my focus is on strengthening methods and performing analyses. Ideally I would obtain results substantial enough to submit to NIPS, EMNLP, or CVPR. At the very least, we aim to publish workshop papers in the aforementioned conferences.

# 5 References

[1] Deeper LSTM and normalized CNN Visual Question Answering model - Jiasen Lu, Xiao Lin, Dhruv Batra, Devi Parikh. Github 2015

[2] Stacked Attention Networks for Image Question Answering - Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Smola. CVPR 2016