Stacking with Auxiliary Features for VQA

Nazneen Fatema Rajani and Raymond J. Mooney
nrjani@cs.utexas.edu and mooney@cs.utexas.edu

Department of Computer Science at the University of Texas at Austin

Introduction

Visual Question Answering (VQA) (Antol et al., 2015) requires both language and image understanding, language grounding capabilities, as well as common-sense knowledge (see Figure 1 for examples). The vision component of a typical VQA system extracts visual features using a deep convolutional neural network (CNN), and the linguistic component encodes the question into a semantic vector using a recurrent neural network (RNN). Most VQA systems have a single underlying method that optimizes a specific loss function and do not leverage the advantage of using multiple diverse models.

The various VQA models have learned to perform well on specific types of questions and images. Therefore, there is an opportunity to combine these models intelligently so as to leverage their diverse strengths.

Auxiliary Features

Question and Answer Types

- Some VQA models are better at certain QA types than others and this information can be used by the stacker at classification time.
- Prefixes of question defined a type, e.g. “What”, “What is”, “What is the”.
- Question type with at least 500 questions and a separate “other” type gave a total of 70 question types which were used as vector of features.
- Infer answer type from the question – “yes/no”, “number” and “other” types.
  - “Does”, “is”, “Was”, “Are”, “Has” classified as yes/no type.
  - “How many”, “what time”, “what number” classified as number type.

Question Features

- Bag-of-Words of the tokens that occur at least 5 times in the questions (on a validation set) are used as features.
- Including a BOW for the question as auxiliary features equips the stacker to efficiently learn which words are important and can aid in classifying answers.

Image Features

- Deep visual features of the VGGNet’s 67 layer contributed 4096 features.
- Using such image features enables the stacker to learn to rely on systems that are good at identifying answers for particular types of images.

Using Explanation as features

- The localization map generated using GradCAM (Selvaraju et al., 2017) by each VQA model serves as a visual explanation for the predicted output of that model (Figure 3).
- We take the absolute gray-scale value of the localization-maps in each of model and rank the pixels according to their spatial attention intensity.
- Then, we compute the correlation between the two ranked lists using the Kendall correlation.

Component VQA systems

We use three diverse individual VQA models as part of our ensemble system:
- LSTM (Antol et al., 2015): This model uses VGGNet to embed the image and two layer LSTM to embed the question. The image and question vectors are fused via element-wise multiplication.
- HieCoAtt (Lu et al., 2016): This model jointly reasons about the visual and language components using two types of “co-attention” – parallel and alternating.
- MCB (Fukui et al., 2016): This model uses the 152-layer ResNet network to embed the image and LSTM to embed the question. The two vectors are combined using the outer product which is made efficient using the multilinear compact bilinear pool (Gao et al., 2016). We used the single system MCB model as a component in our ensemble.

The top performing VQA system in the 2016 competition was an ensemble of 7 MCB models. Their model is pre-trained on the VQA training dataset and they concatenate learned word embedding with pre-trained GloVe vectors.

Results

- We used a neural network as the meta-classifier implemented in Keras.
- We found that using late fusion for combining auxiliary features worked better.

Table 1: Accuracy results on the VQA test-standard set. The first block shows performance of a VQA model that uses external data for pre-training, the second block shows single system VQA models, the third block shows ensemble VQA models.

<table>
<thead>
<tr>
<th>Method</th>
<th>All</th>
<th>Yes/No</th>
<th>Number</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPPNet (Noh et al., 2016)</td>
<td>65.36</td>
<td>80.28</td>
<td>36.02</td>
<td>42.24</td>
</tr>
<tr>
<td>b i OWMG (Zhong et al., 2015)</td>
<td>58.72</td>
<td>76.85</td>
<td>33.03</td>
<td>42.62</td>
</tr>
<tr>
<td>N M Ns (Andreas et al., 2016)</td>
<td>58.70</td>
<td>81.20</td>
<td>37.70</td>
<td>44.00</td>
</tr>
<tr>
<td>L S T M (Antol et al. 2015)</td>
<td>58.20</td>
<td>80.60</td>
<td>36.50</td>
<td>43.70</td>
</tr>
<tr>
<td>H i e C o A t t (Lu et al., 2016)</td>
<td>61.80</td>
<td>79.70</td>
<td>38.70</td>
<td>51.70</td>
</tr>
<tr>
<td>M C B (Single system) (Fukui et al., 2016)</td>
<td>62.56</td>
<td>80.68</td>
<td>35.59</td>
<td>52.93</td>
</tr>
<tr>
<td>M C B (Ensemble) (Fukui et al., 2016)</td>
<td>66.50</td>
<td>83.20</td>
<td>39.50</td>
<td>58.00</td>
</tr>
<tr>
<td>V o t i n g (M C B + H i e C o A t t + L S T M)</td>
<td>60.31</td>
<td>80.22</td>
<td>34.92</td>
<td>48.83</td>
</tr>
<tr>
<td>Stacking</td>
<td>63.12</td>
<td>81.61</td>
<td>36.07</td>
<td>53.77</td>
</tr>
<tr>
<td>+ Q A type features</td>
<td>65.25</td>
<td>82.01</td>
<td>36.50</td>
<td>57.15</td>
</tr>
<tr>
<td>+ Question features</td>
<td>65.50</td>
<td>82.26</td>
<td>37.21</td>
<td>57.35</td>
</tr>
<tr>
<td>+ Image features</td>
<td>65.54</td>
<td>82.28</td>
<td>38.63</td>
<td>57.32</td>
</tr>
<tr>
<td>+ Explanation features</td>
<td>67.26</td>
<td>82.62</td>
<td>39.00</td>
<td>58.34</td>
</tr>
</tbody>
</table>

References and Acknowledgements

- Zhengqing Cai, Jiasen Yang, Jacob Andreas, Alan g. Miller, and Devi Parikh. A deep learning approach to question answering with visual and dialogue context. CVPR 2016.

This research was supported by the DARPA XAI grant.