Bottom-up and Top-Down Attention for Image Captioning and Visual Question Answering

Jiyang Zhang, Tong Gao
Background

• **Image captioning** and **visual question answering** are problems combining image and language understanding.

• To solve these problems, it is often necessary to perform visual processing, or even reasoning to generate high quality outputs.

• Most conventional visual attention mechanisms are of the **top-down** variety: Given context, model attend to one or more layers of CNN.
Problem

- CNN processes input regions in a uniform grid space, regardless of the content of the image
- Attention on grid space - only on partial object
Our Model

- **Top-down mechanism**: use task-specific context to predict an attention distribution over the image
- **Bottom-up mechanism**: use Faster R-CNN to propose a set of salient image regions
Advantages

• With Faster R-CNN, the model attends to the **full object** now.

• We are able to pre-train it on object detection datasets, leveraging cross-domain knowledge.
Overview

- Bottom-up Attention Model
- Top-down Attention Model
  - Captioning Model
  - VQA Model
- Datasets
- Results
- Conclusion
- Critique
- Discussion
Bottom-up Attention Model

Object category

Regions of Interest (RoIs) from a proposal method

“Backbone” network: AlexNet, VGG, ResNet, etc

ConvNet

CNN

Linear + softmax

Linear

Box offset

Per-Region Network

Crop + Resize features

“conv5” features

Run whole image through ConvNet

Input image

Girshick, “Fast R-CNN”, ICCV 2015. Figure copyright Ross Girshick, 2015; source: Reproduced with permission.
Bottom-up Attention Model

**Faster R-CNN:**
Make CNN do proposals!

Jointly train with 4 losses:
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

---

Figure copyright 2015, Ross Girshick; reproduced with permission
Bottom-up Attention Model

**Faster R-CNN:**
Make CNN do proposals!

Jointly train with 4 losses:
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates
5. Final classification score (attributes)

Figure copyright 2015, Ross Girshick; reproduced with permission
Captioning Model (Attention LSTM)

\[ v = \frac{1}{k} \sum_{i} v_i \]

\( \Pi_t \) is a one-hot vector for an input at time step \( t \)
Captioning Model (Attention LSTM)

For each of $k$ image features $v_i$, compute the attention weight $\alpha_{i,t}$:

$$\alpha_{i,t} = w_a^T \tanh(W_{va}v_i + W_{ha}h^1_t)$$

$$\alpha_t = \text{softmax}(\alpha_t)$$

Where $w_a^T$, $W_{va}$ and $W_{ha}$ are learnable parameters. Then,

$$\hat{v}_t = \sum_{i=1}^{k} \alpha_{i,t}v_i$$
Captioning Model (Attention LSTM)

\[ x_t^2 = [v_t, h_t^1] \]
Objective

1. Given target ground truth sequence $y_{1:T}^*$, minimize the cross entropy loss:

$$L_{XE}(\theta) = -\sum_{t=1}^{T} \log(p_{\theta}(y_t^* \mid y_{1:t-1}^*))$$

2. After training, spend some time optimizing the model for CIDEr score. Minimize:

$$L_R(\theta) = -\mathbb{E}_{y_{1:T} \sim p_\theta} [r(y_{1:T})]$$

where $r$ is the score function.
Objective

Minimize: \[ L_R(\theta) = -E_{y_{1:T} \sim p_\theta}[r(y_{1:T})] \]

Following **Self-Critical Sequence Training (SCST)**, the gradient can be approximated as:

\[
\nabla_\theta L_R(\theta) \approx -(r(y^s_{1:T}) - r(\hat{y}_{1:T})) \nabla_\theta \log p_\theta(y^s_{1:T})
\]

Where \( y^s_{1:T} \) is sampled caption (sample from decoded beam) and \( r(\hat{y}_{1:T}) \) is the baseline score obtained from greedily decoding the current model.
VQA Model

Defines non-linear transformation \( f: x \in \mathbb{R}^m \rightarrow y \in \mathbb{R}^n \) with parameters \( \{W, W', b, b'\} \)

\[
\begin{align*}
\tilde{y} &= \tanh(Wx + b) \\
g &= \sigma(W'x + b') \\
y &= \tilde{y} \circ g
\end{align*}
\]

Element-wise product of two parallel non-linear network, activated by tanh and \( \sigma \) respectively.
VQA Model

Compute $\alpha$ (attention score)

\[ \tilde{v}_t = \sum_{i=1}^{K} \alpha_{i,t} v_i \]
VQA Model

\[ h = f_q(q) \circ f_v(\hat{v}) \]

Where \( h \) is the joint representation of words and images

\[ p(y) = \sigma(W_0f_o(h)) \]

Confidence score for every candidate answers, trained with binary cross entropy loss
Dataset

- **Visual Genome dataset**
  - pretrain bottom-up attention model
  - the dataset contains 108K images densely annotated, containing objects, attributes and relationships, and visual question answers
  - ensure that any images found in both datasets are contained in the same split
  - augment VQA v2.0 training data

- **Microsoft COCO Dataset**
  - Image caption task

- **VQA v2.0 Dataset**
  - Visual Question Answering task
  - attempts to minimize the effectiveness of learning dataset priors by balancing the answers to each question
ResNet Baseline

- To quantify the impact of bottom-up attention
- Uses a ResNet CNN pretrained on ImageNet to encode each image in place of the bottom-up attention
- Image caption: use the final convolutional layer of Resnet-101, resize the output to a fixed size spatial representation of 10×10
- VQA: varying the size of output representations, 14×14, 7×7, 1×1
## Image caption results

<table>
<thead>
<tr>
<th></th>
<th>Cross-Entropy Loss</th>
<th>CIDEr Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU-1  BLEU-4  METEOR  ROUGE-L  CIDEr  SPICE</td>
<td>BLEU-1  BLEU-4  METEOR  ROUGE-L  CIDEr  SPICE</td>
</tr>
<tr>
<td>SCST:Att2in [34]</td>
<td>- 31.3  26.0  54.3  101.3  -</td>
<td>- 33.3  26.3  55.3  111.4  -</td>
</tr>
<tr>
<td>SCST:Att2all [34]</td>
<td>- 30.0  25.9  53.4  99.4  -</td>
<td>- 34.2  26.7  55.7  114.0  -</td>
</tr>
<tr>
<td>Ours: ResNet</td>
<td>74.5 33.4 26.1 54.4 105.4 19.2</td>
<td>76.6 34.0 26.5 54.9 111.1 20.2</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td>77.2 36.2 27.0 56.4 113.5 20.3</td>
<td>79.8 36.3 27.7 56.9 120.1 21.4</td>
</tr>
<tr>
<td>Relative Improvement</td>
<td>4% 8% 3% 4% 8% 6%</td>
<td>4% 7% 5% 4% 8% 6%</td>
</tr>
<tr>
<td></td>
<td>Cross-Entropy Loss</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td></td>
<td>SPICE</td>
<td>Objects</td>
</tr>
<tr>
<td>Ours: ResNet</td>
<td>19.2</td>
<td>35.4</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td><strong>20.3</strong></td>
<td><strong>37.1</strong></td>
</tr>
</tbody>
</table>

Table 2. Breakdown of SPICE F-scores over various subcategories on the MSCOCO Karpathy test split. Our Up-Down model outperforms the ResNet baseline at identifying objects, as well as detecting object attributes and the relations between objects.
dependency parse trees

A young girl standing on top of a tennis court

semantic scene graph
<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU-1 c5</th>
<th>BLEU-1 c40</th>
<th>BLEU-2 c5</th>
<th>BLEU-2 c40</th>
<th>BLEU-3 c5</th>
<th>BLEU-3 c40</th>
<th>BLEU-4 c5</th>
<th>BLEU-4 c40</th>
<th>METEOR c5</th>
<th>METEOR c40</th>
<th>ROUGE-L c5</th>
<th>ROUGE-L c40</th>
<th>CIDEr c5</th>
<th>CIDEr c40</th>
<th>SPICE c5</th>
<th>SPICE c40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Review Net [48]</td>
<td>72.0</td>
<td>90.0</td>
<td>55.0</td>
<td>81.2</td>
<td>41.4</td>
<td>70.5</td>
<td>31.3</td>
<td>59.7</td>
<td>25.6</td>
<td>34.7</td>
<td>53.3</td>
<td>68.6</td>
<td>96.5</td>
<td>96.9</td>
<td>18.5</td>
<td>64.9</td>
</tr>
<tr>
<td>Adaptive [27]</td>
<td>74.8</td>
<td>92.0</td>
<td>58.4</td>
<td>84.5</td>
<td>44.4</td>
<td>74.4</td>
<td>33.6</td>
<td>63.7</td>
<td>26.4</td>
<td>35.9</td>
<td>55.0</td>
<td>70.5</td>
<td>104.2</td>
<td>105.9</td>
<td>19.7</td>
<td>67.3</td>
</tr>
<tr>
<td>PG-BCMR [24]</td>
<td>75.4</td>
<td>-</td>
<td>59.1</td>
<td>-</td>
<td>44.5</td>
<td>-</td>
<td>33.2</td>
<td>-</td>
<td>25.7</td>
<td>-</td>
<td>55.0</td>
<td>-</td>
<td>101.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>SCST:Att2all [34]</td>
<td>78.1</td>
<td>93.7</td>
<td>61.9</td>
<td>86.0</td>
<td>47.0</td>
<td>75.9</td>
<td>35.2</td>
<td>64.5</td>
<td>27.0</td>
<td>35.5</td>
<td>56.3</td>
<td>70.7</td>
<td>114.7</td>
<td>116.7</td>
<td>20.7</td>
<td>68.9</td>
</tr>
<tr>
<td>LSTM-A3 [49]</td>
<td>78.7</td>
<td>93.7</td>
<td>62.7</td>
<td>86.7</td>
<td>47.6</td>
<td>76.5</td>
<td>35.6</td>
<td>65.2</td>
<td>27.0</td>
<td>35.4</td>
<td>56.4</td>
<td>70.5</td>
<td>116.0</td>
<td>118.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td><strong>80.2</strong></td>
<td><strong>95.2</strong></td>
<td><strong>64.1</strong></td>
<td><strong>88.8</strong></td>
<td><strong>49.1</strong></td>
<td><strong>79.4</strong></td>
<td><strong>36.9</strong></td>
<td><strong>68.5</strong></td>
<td><strong>27.6</strong></td>
<td><strong>36.7</strong></td>
<td><strong>57.1</strong></td>
<td><strong>72.4</strong></td>
<td><strong>117.9</strong></td>
<td><strong>120.5</strong></td>
<td><strong>21.5</strong></td>
<td><strong>71.5</strong></td>
</tr>
</tbody>
</table>

Table 3. Highest ranking published image captioning results on the online MSCOCO test server. Our submission, an ensemble of 4 models optimized for CIDEr with different initializations, outperforms previously published work on all reported metrics. At the time of submission (18 July 2017), we also outperformed all unpublished test server submissions.
VQA Results

<table>
<thead>
<tr>
<th></th>
<th>Yes/No</th>
<th>Number</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours: ResNet (1×1)</td>
<td>76.0</td>
<td>36.5</td>
<td>46.8</td>
<td>56.3</td>
</tr>
<tr>
<td>Ours: ResNet (14×14)</td>
<td>76.6</td>
<td>36.2</td>
<td>49.5</td>
<td>57.9</td>
</tr>
<tr>
<td>Ours: ResNet (7×7)</td>
<td>77.6</td>
<td>37.7</td>
<td>51.5</td>
<td>59.4</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td>80.3</td>
<td>42.8</td>
<td>55.8</td>
<td>63.2</td>
</tr>
<tr>
<td>Relative Improvement</td>
<td>3%</td>
<td>14%</td>
<td>8%</td>
<td>6%</td>
</tr>
</tbody>
</table>

Table 4. Single-model performance on the VQA v2.0 validation set. The use of bottom-up attention in the Up-Down model provides a significant improvement over the best ResNet baseline across all question types, even though the ResNet baselines use almost twice as many convolutional layers.
# VQA Results

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Yes/No</th>
<th>Number</th>
<th>Other</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior [12]</td>
<td>61.20</td>
<td>0.36</td>
<td>1.17</td>
<td>25.98</td>
</tr>
<tr>
<td>Language-only [12]</td>
<td>67.01</td>
<td>31.55</td>
<td>27.37</td>
<td>44.26</td>
</tr>
<tr>
<td>d-LSTM+n-I [26, 12]</td>
<td>73.46</td>
<td>35.18</td>
<td>41.83</td>
<td>54.22</td>
</tr>
<tr>
<td>MCB [11, 12]</td>
<td>78.82</td>
<td>38.28</td>
<td>53.36</td>
<td>62.27</td>
</tr>
<tr>
<td>UPMC-LIP6</td>
<td>82.07</td>
<td>41.06</td>
<td>57.12</td>
<td>65.71</td>
</tr>
<tr>
<td>Athena</td>
<td>82.50</td>
<td>44.19</td>
<td>59.97</td>
<td>67.59</td>
</tr>
<tr>
<td>HDU-USYD-UNCC</td>
<td>84.50</td>
<td>45.39</td>
<td>59.01</td>
<td>68.09</td>
</tr>
<tr>
<td>Ours: Up-Down</td>
<td>86.60</td>
<td>48.64</td>
<td>61.15</td>
<td>70.34</td>
</tr>
</tbody>
</table>

Table 5. VQA v2.0 test-standard server accuracy as at 8 August 2017, ranking our submission against published and unpublished work for each question type. Our approach, an ensemble of 30 models, outperforms all other leaderboard entries.
Two men playing frisbee in a dark field.
Errors

Critique

• Randomly initialized word embedding in image captioning task, but GloVe vectors on VQA model?
• Why don't merge overlapping classes when processing Visual Genome Dataset?
  - Perform stemming to reduce the class size (e.g. trees->tree)
  - Use WordNet to merge synonyms
• The model submitted to VQA challenge is trained with additional Q&A from Visual Genome - cheating?
• Also - they use 30 ensembled models on the test evaluation server?
• Their image captioning model forces the decoder to generate unique words in a row, but some prepositions can appear for twice or more
  - only filter nouns
Critique

- Curious about the length of image features with relation to the performance. Will it be harder to generate captions for more complicated images.
- Evaluation only includes automatic metrics, needs more human evaluation in image caption generation task, like relevance, expressiveness, concreteness, creativity.
- Need analysis of results of different types of questions, e.g. “Is the” or “what is” questions. And it will be interesting to show the distribution of age of questions for different levels of accuracies achieved by our system, estimate the model can perform as well as humans in which age.
- Other things could try:
- Is it possible to also apply attention to words in the question for VQA?
Thank you!
Non-maximum Suppression

Given a list of bounding box \( L = [b_1, b_2, \ldots] \)

1. Pick the bounding box \( b_h \) with highest confidence score
   - Remove \( b_h \) from \( L \)
   - Append \( b_h \) to the output list
2. Compute the IoU (intersection-over-union) between \( b_h \) over all other boxes in \( L \)
3. Remove boxes from \( L \) with IoU > threshold
4. Repeat until \( L \) becomes empty

\[
\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}
\]
Why Sigmoid?

- The sigmoid outputs allow optimization for multiple correct answers per question (occasional cases in VQA)

- Each answer in VQA are labeled with soft accuracies in [0,1]

- Thus, soft scores as targets provides a slightly richer training signal than binary targets
What is SPICE?

- (a) A young girl standing on top of a tennis court.
- (b) A giraffe standing on top of a green field.

High n-gram similarity

- (c) A shiny metal pot filled with some diced veggies.
- (d) The pan on the stove has chopped vegetables in it.

Low n-gram similarity