Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Xu et al. (2016)

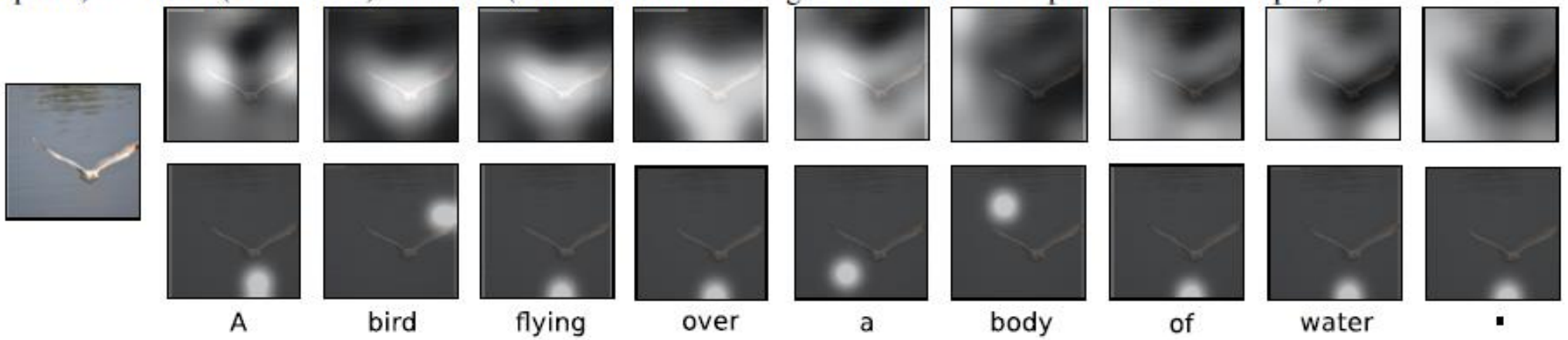
Presented by Benjamin Striner, 9/19/2017

Goals

What is the purpose of this model?

Model “looks” at image while generating

Figure 2. Attention over time. As the model generates each word, its attention changes to reflect the relevant parts of the image. “soft” (top row) vs “hard” (bottom row) attention. (Note that both models generated the same captions in this example.)



Each word corresponds to a location

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

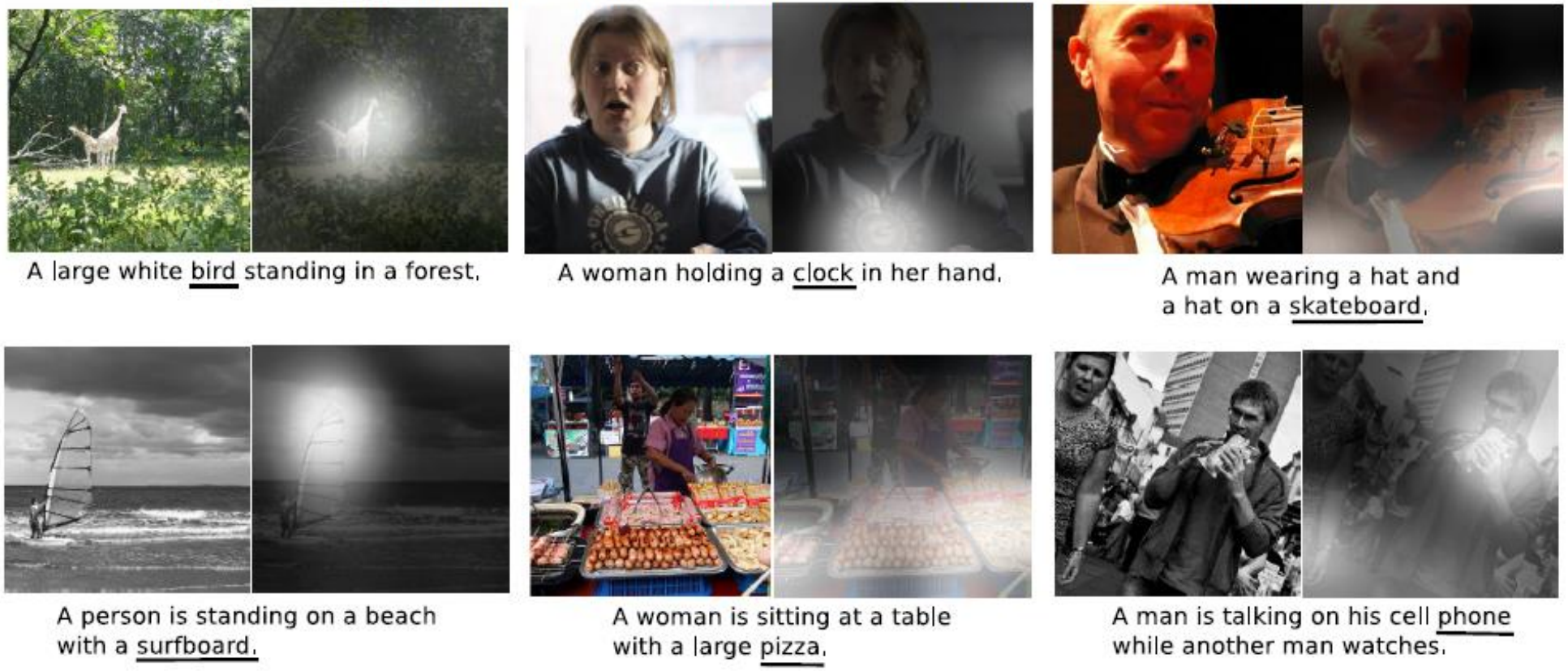
A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Locations help understand mistakes

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.



Model Overview

How do you learn attention?

Encoder-Decoder Model

- CNN for encoding
- RNN with attention for decoding

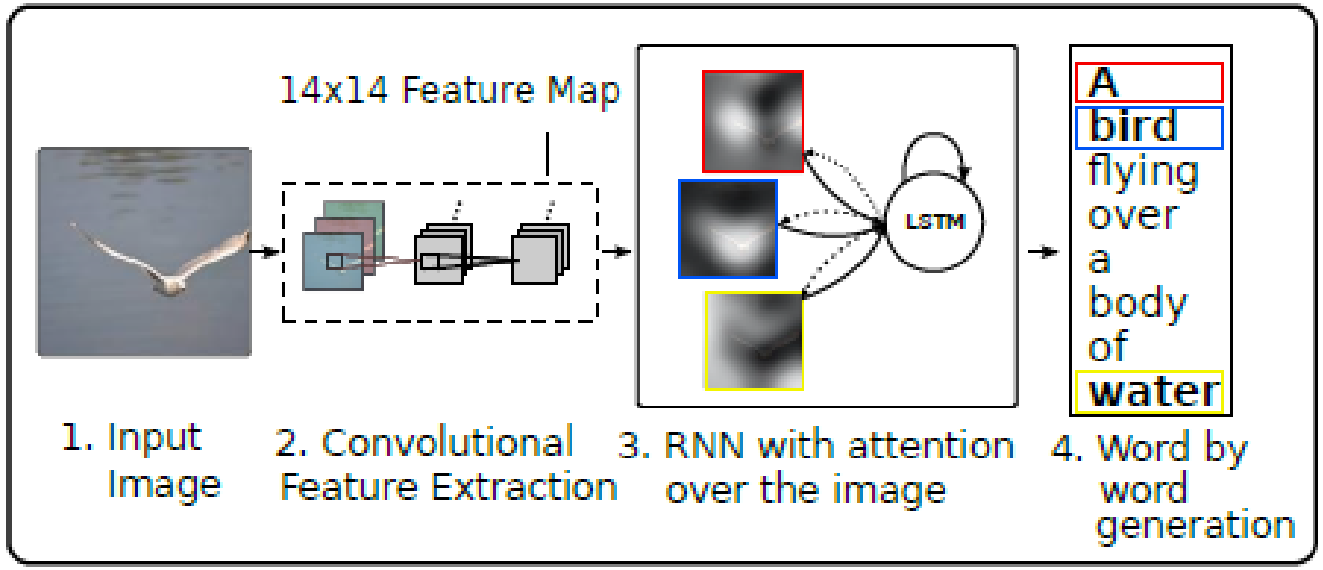


Image Encoder

- CNN to extract a set of feature vectors
  - Correspond to locations in the image
  - Termed “annotation vectors”
- Use pretrained VGGnet
  - Use fourth from last layer (14x14)
  - No finetuning

$$a = \{\mathbf{a}_1, \ldots, \mathbf{a}_L\}, \ \mathbf{a}_i \in \mathbb{R}^D$$

Decoder LSTM

- Standard LSTM trained to output captions
  - E: an embedding matrix
  - y: output word
  - h: hidden state
  - z: context vector
- Initial state output by MLP

$$\mathbf{c}_0 = f_{\text{init,c}}(\frac{1}{L}\sum_i^L \mathbf{a}_i)$$

$$\mathbf{h}_0 = f_{\text{init,h}}(\frac{1}{L}\sum_i^L \mathbf{a}_i)$$

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T_{D+m+n,n} \begin{pmatrix} \mathbf{E}\mathbf{y}_{t-1} \\ \mathbf{h}_{t-1} \\ \hat{\mathbf{z}}_t \end{pmatrix} \quad (1)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (2)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t). \quad (3)$$

Decoder Output

- Output layer predicts next word based on LSTM hidden state, context vector, and previous word

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)) \quad (7)$$

Context Attention

- Attention function is an MLP
  - Inputs are annotations and hidden state
  - Output uses softmax
- Context vector is a function of annotation vectors and attention

$$e_{ti} = f_{\text{att}}(\mathbf{a}_i, \mathbf{h}_{t-1}) \tag{4}$$

$$\alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{k=1}^{L} \exp(e_{tk})}. \tag{5}$$

$$\hat{\mathbf{z}}_t = \phi\left(\{\mathbf{a}_i\}, \{\alpha_i\}\right), \tag{6}$$

Hard Attention Mechanism

Stochastic choice of location for attention

Hard Attention

- Attention is a stochastic choice of a single location
- Multiply attention by annotation to get context vector

$$p(s_{t,i} = 1 \mid s_{j<t}, \mathbf{a}) = \alpha_{t,i} \tag{8}$$

$$\hat{\mathbf{z}}_t = \sum_i s_{t,i} \mathbf{a}_i. \tag{9}$$

Model Objective

- Model attempts to maximize the probability of generating target words given an attribute vector
- Calculate lower bound on log(p(y|a))

$$
\begin{aligned}
L_s &= \sum_s p(s \mid \mathbf{a}) \log p(\mathbf{y} \mid s, \mathbf{a}) \\
&\leq \log \sum_s p(s \mid \mathbf{a}) p(\mathbf{y} \mid s, \mathbf{a}) \\
&= \log p(\mathbf{y} \mid \mathbf{a}) \qquad (10)
\end{aligned}
$$

$$
\frac{\partial L_s}{\partial W} = \sum_s p(s \mid \mathbf{a}) \left[\frac{\partial \log p(\mathbf{y} \mid s, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid s, \mathbf{a}) \frac{\partial \log p(s \mid \mathbf{a})}{\partial W} \right]. \qquad (11)
$$

Monte Carlo Sampling

- Previous formula can be approximated by sampling

$$\tilde{s}_t \sim \text{Multinoulli}_L(\{\alpha_i\})$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \Big[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) \frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} \Big] \quad (12)$$

Monte Carlo Details

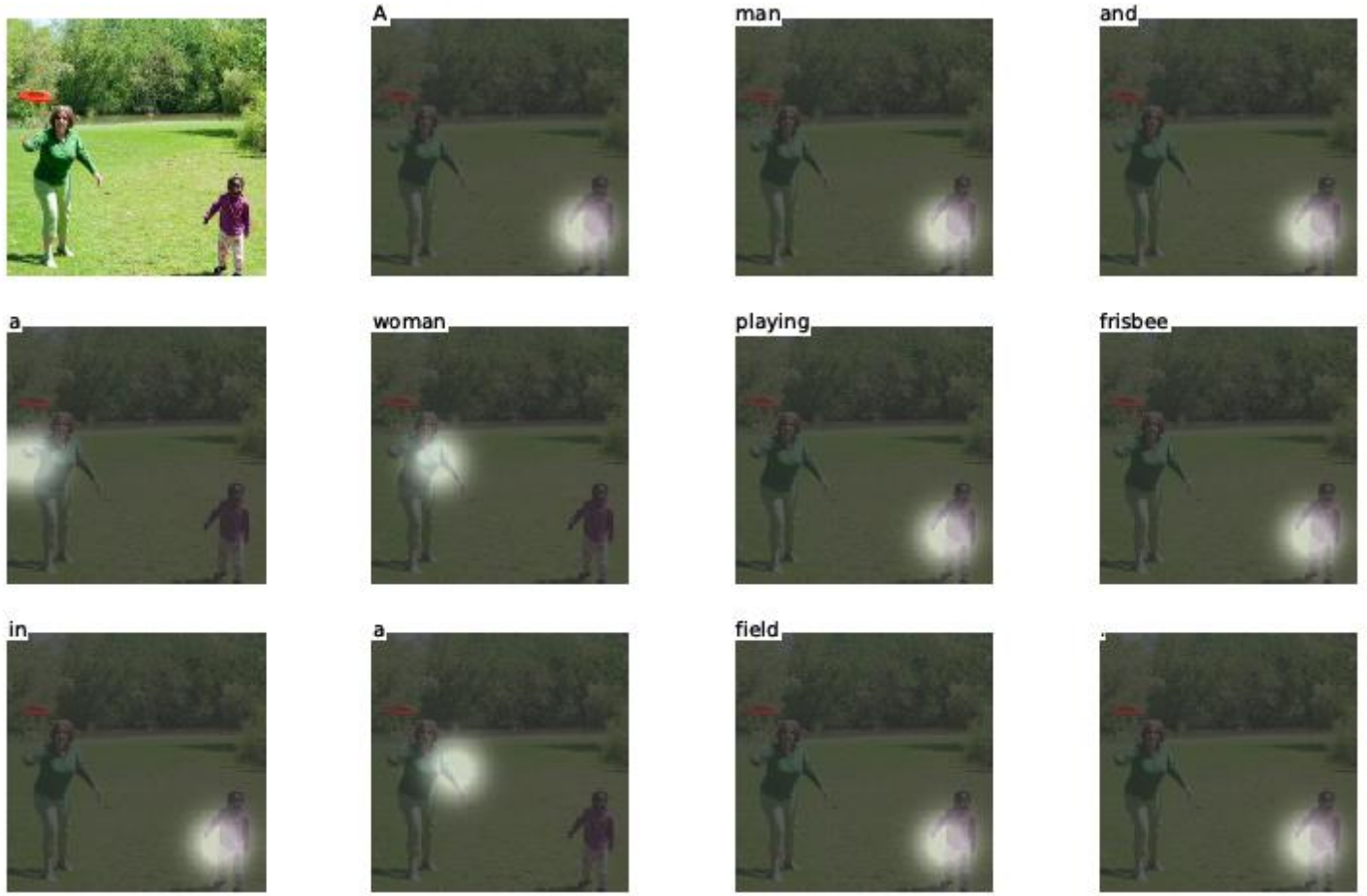
- Add a moving average term
- Add an entropy term
- With probability 0.5, set sampled location to its expected value

$$b_k = 0.9 \times b_{k-1} + 0.1 \times \log p(\mathbf{y} \mid \tilde{s}_k, \mathbf{a})$$

$$\frac{\partial L_s}{\partial W} \approx \frac{1}{N} \sum_{n=1}^{N} \left[\frac{\partial \log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a})}{\partial W} + \right.$$

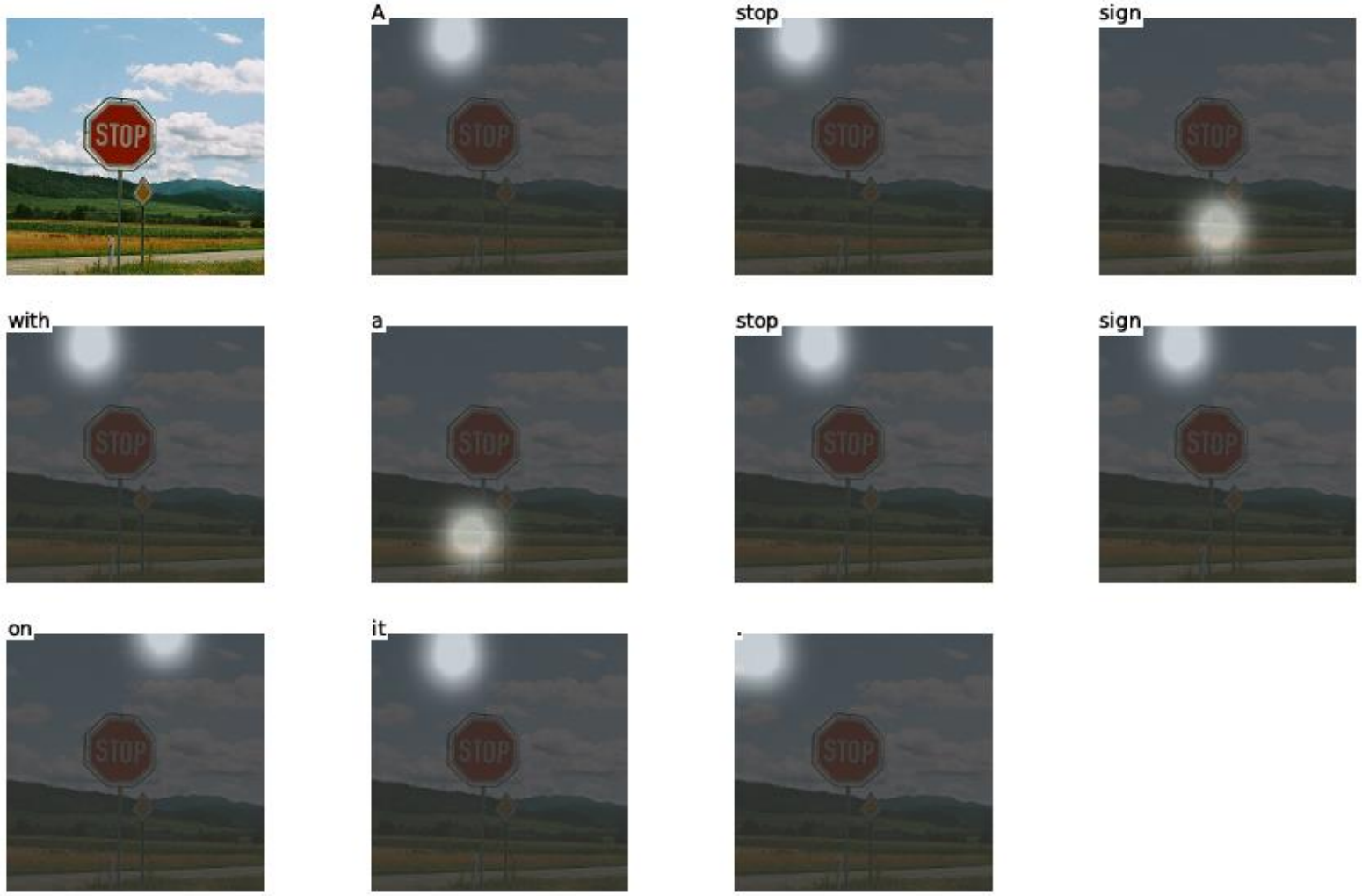
$$\left. \lambda_r (\log p(\mathbf{y} \mid \tilde{s}^n, \mathbf{a}) - b) \frac{\partial \log p(\tilde{s}^n \mid \mathbf{a})}{\partial W} + \lambda_e \frac{\partial H[\tilde{s}^n]}{\partial W} \right]$$

Hard Attention Example 1



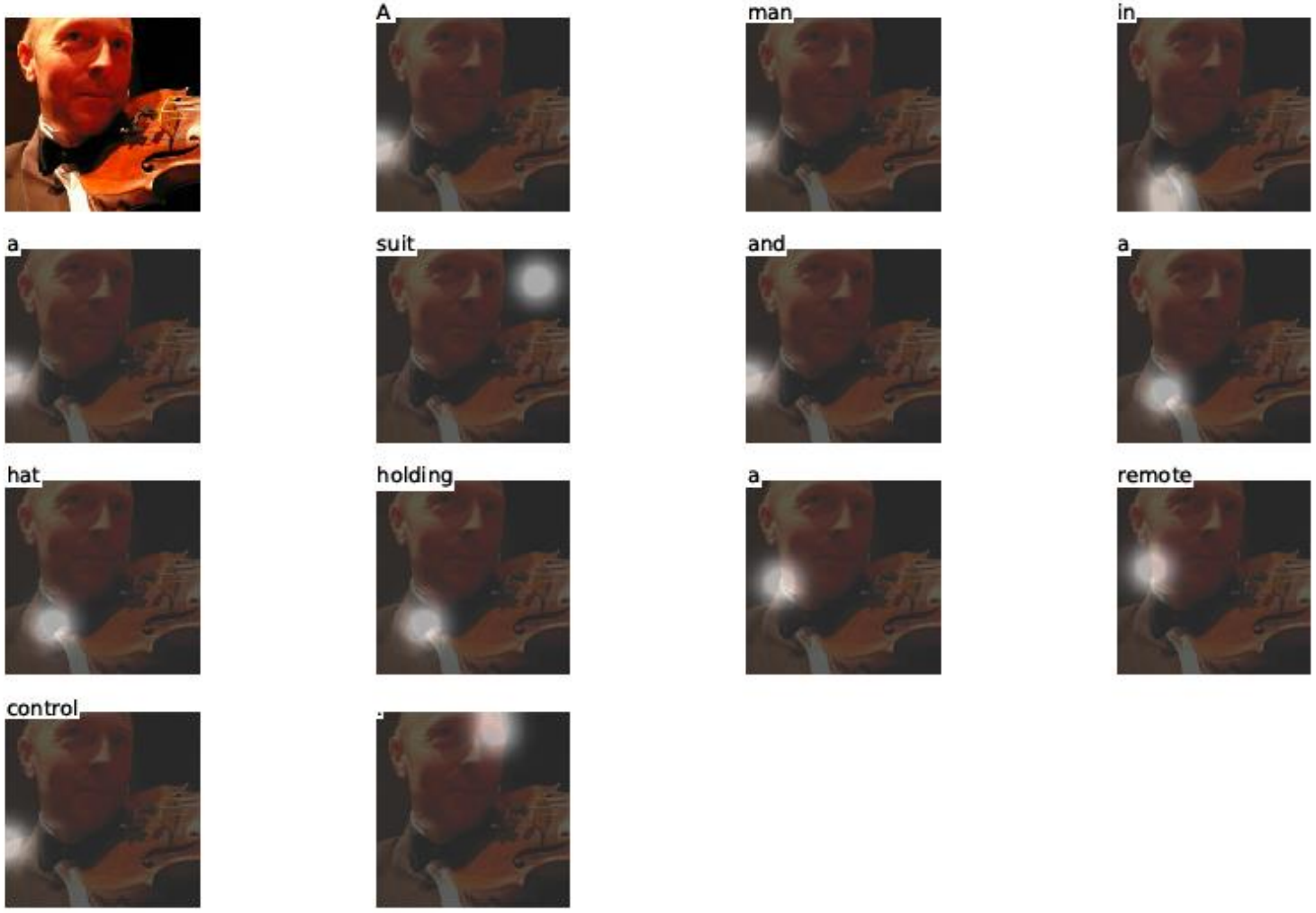
(a) A man and a woman playing frisbee in a field.

Hard Attention Example 2



(a) A stop sign with a stop sign on it.

Hard Attention Example 3



(a) A man in a suit and a hat holding a remote control.

Soft Attention Mechanism

Differentiable attention over entire image

Soft Attention

- The expectation of the context is just a sum product
- Pass expectation to LSTM instead of a sample
- Much easier than sampling

$$\mathbb{E}_{p(s_t|a)}[\hat{\mathbf{z}}_t] = \sum_{i=1}^{L} \alpha_{t,i}\mathbf{a}_i \qquad (13)$$

Gating Mechanism

- Add gating mechanism on top of attention

$$\beta_t = \sigma(f_\beta(\mathbf{h}_{t-1}))$$

$$\phi(\{\mathbf{a}_i\}, \{\alpha_i\}) = \beta \sum_i^L \alpha_i \mathbf{a}_i$$

Doubly Stochastic Attention

- Penalize model so it tends to use the entire image
- Regularize by squared difference from 1

$$\sum_i \alpha_{ti} = 1 \quad \sum_t \alpha_{ti} \approx 1$$

$$L_d = -\log(P(\mathbf{y}|\mathbf{x})) + \lambda \sum_i^L (1 - \sum_t^C \alpha_{ti})^2 \qquad (14)$$

Why does this work?

- Expected hidden state of LSTM is approximately hidden state given expected input
  - N(t,k,i): (timestep, word, location)

$$p(\mathbf{y}_t|\mathbf{a}, \mathbf{y}_1^{t-1}) \propto \exp(\mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)) \quad (7)$$

$$\mathbf{n}_t = \mathbf{L}_o(\mathbf{E}\mathbf{y}_{t-1} + \mathbf{L}_h\mathbf{h}_t + \mathbf{L}_z\hat{\mathbf{z}}_t)$$

$$NWGM[p(y_t = k \mid \mathbf{a})] = \frac{\prod_i \exp(n_{t,k,i})^{p(s_{t,i}=1|a)}}{\sum_j \prod_i \exp(n_{t,j,i})^{p(s_{t,i}=1|a)}}$$

$$= \frac{\exp(\mathbb{E}_{p(s_t|a)}[n_{t,k}])}{\sum_j \exp(\mathbb{E}_{p(s_t|a)}[n_{t,j}])}$$

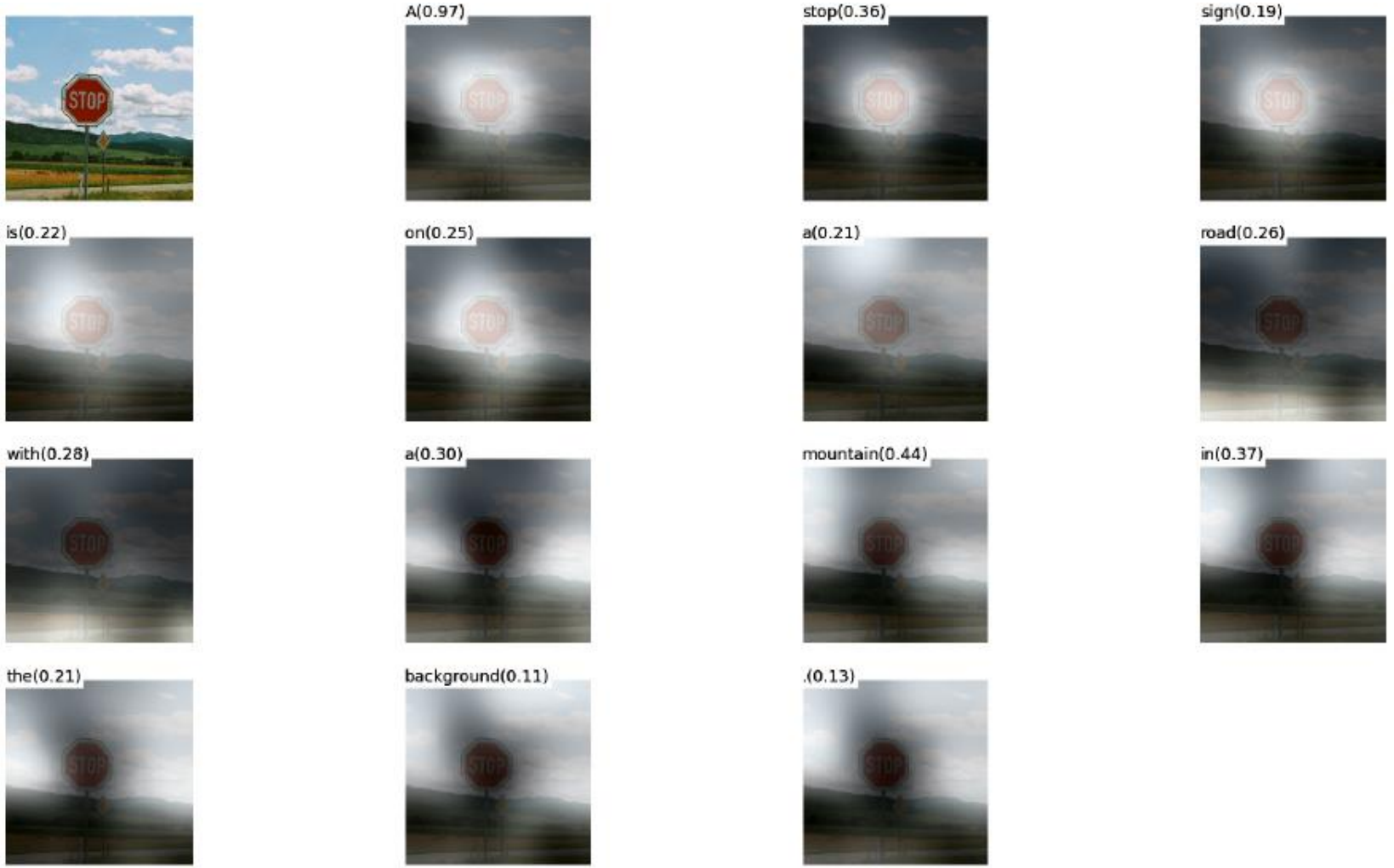
$$NWGM[p(y_t = k \mid \mathbf{a})] \approx \mathbb{E}[p(y_t = k \mid \mathbf{a})]$$

Soft Attention Example 1

(b) A woman is throwing a frisbee in a park.

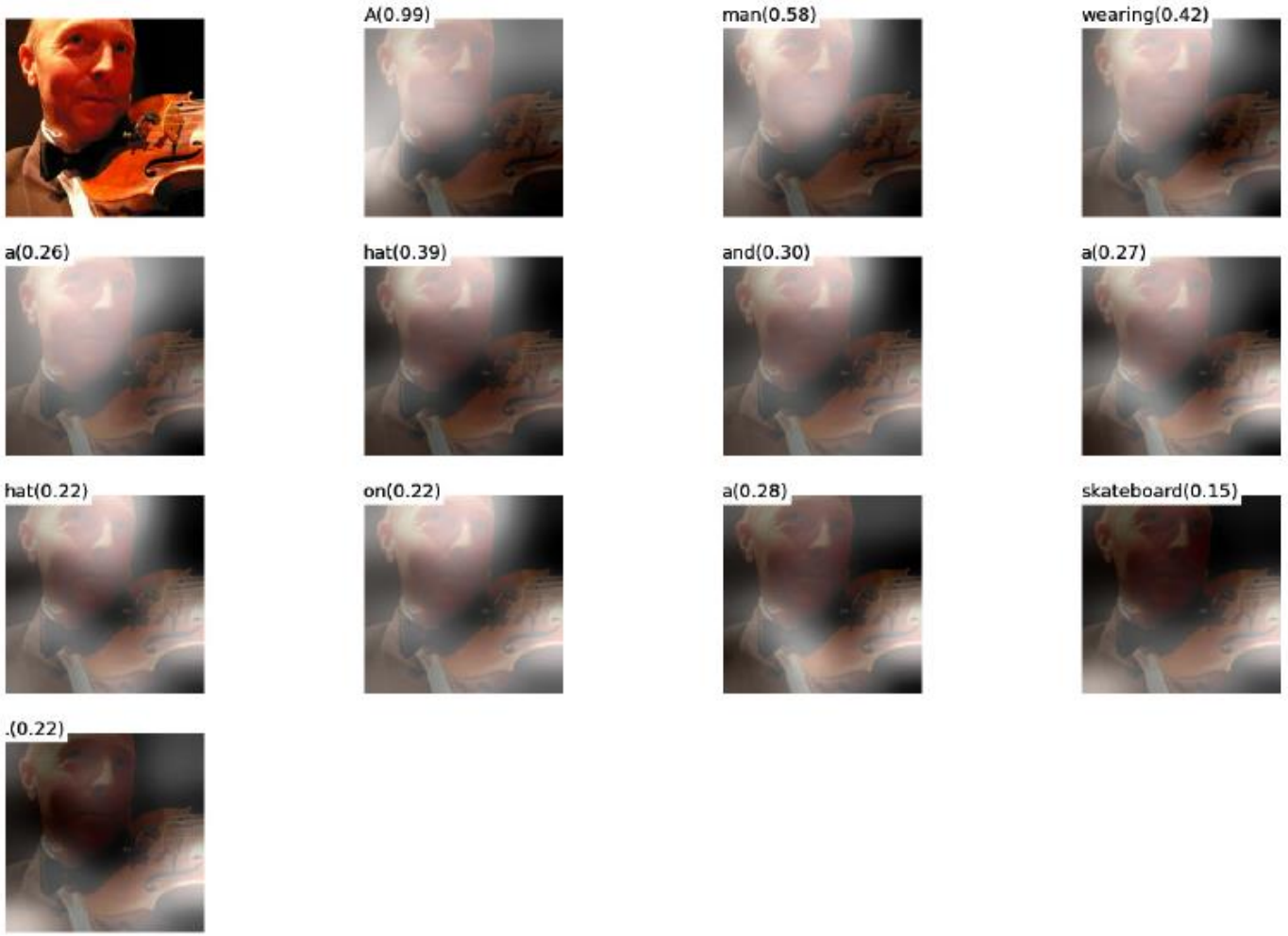
Figure 6.

Soft Attention Example 2



(b) A stop sign is on a road with a mountain in the background.

Soft Attention Example 3



(b) A man wearing a hat and a hat on a skateboard.

Experiments

What can the model do?

Datasets

- Flikr
  - RMSprop for Flikr8k
  - Adam for Flikr30k
- MSCOCO
  - Adam
  - Tokenized for consistency with Flikr

Training Details

- Batch samples by sentence length to increase efficiency
- Dropout
- Early stopping
- Whetlab to optimize parameters

Annotation Results

- Annotate images and report BLEU and METEOR

Table 1. BLEU-1,2,3,4/METEOR metrics compared to other methods, † indicates a different split, (—) indicates an unknown metric, ∘ indicates the authors kindly provided missing metrics by personal communication, Σ indicates an ensemble, *a* indicates using AlexNet

		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

Discussion

Conclusions

- Attention can improve on state-of-the-art results
- Attention can provide interpretability and explainability
- Want to encourage work on visual attention
- Encoder-decoder with attention can be used in other domains
- Does not require object detection or localization training

Questions?