Deep Visual-Semantic Alignments for Generating Image Descriptions

Karpathy, A and Fei-Fei, L (2015)

Presented by Benjamin Striner

9/19/2017

Goal and Graphics

What are they trying to build?

Concept Art

- Goals are beautifully illustrated
- (not actual model output)

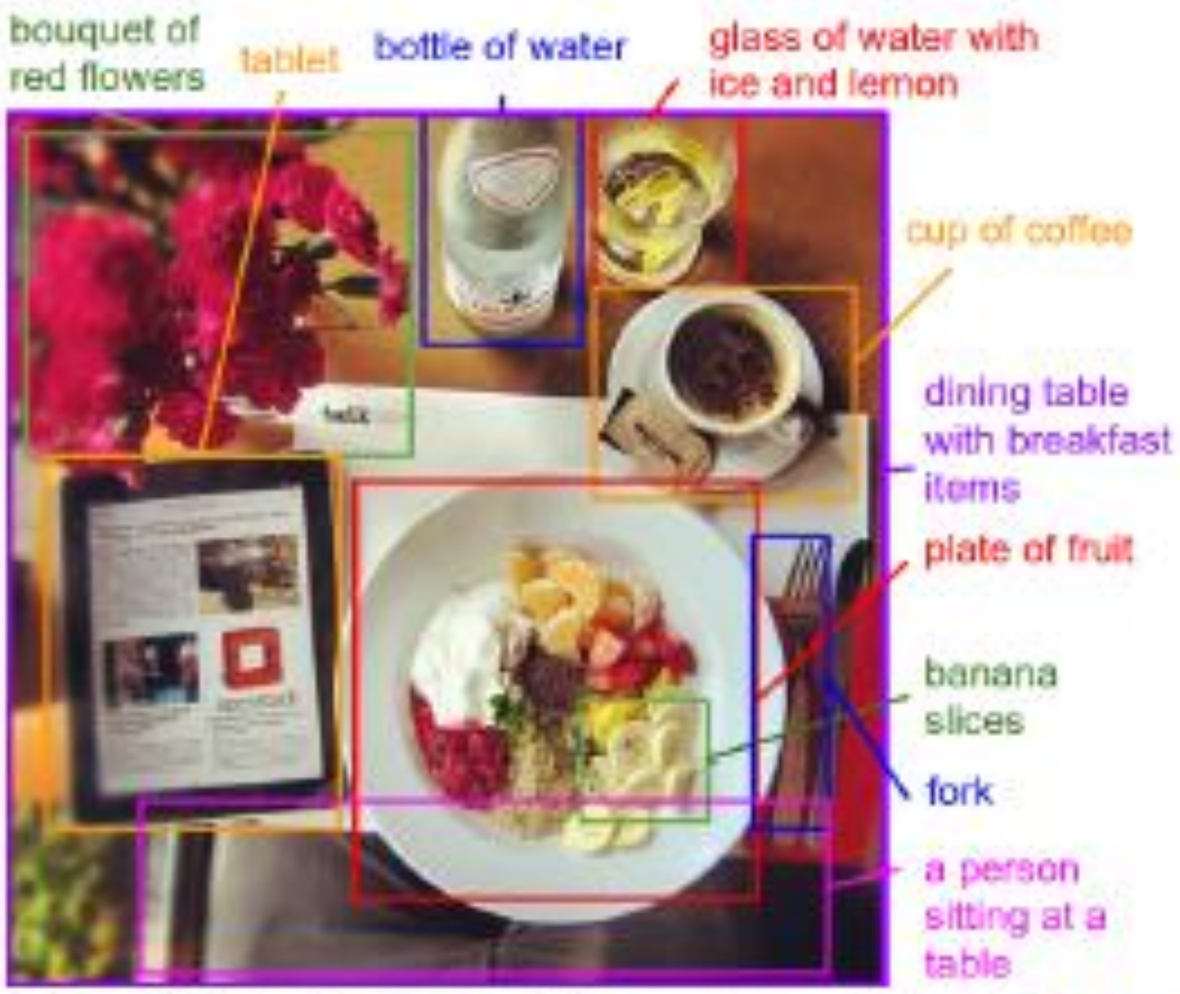

Figure 1. Motivation/Concept Figure: Our model treats language as a rich label space and generates descriptions of image regions.

Tasks

- Several related tasks
  - Image classification
  - Object detection
  - Image and annotation alignment
  - Whole image annotation generation
  - Image region annotation generation

Predicted Alignments

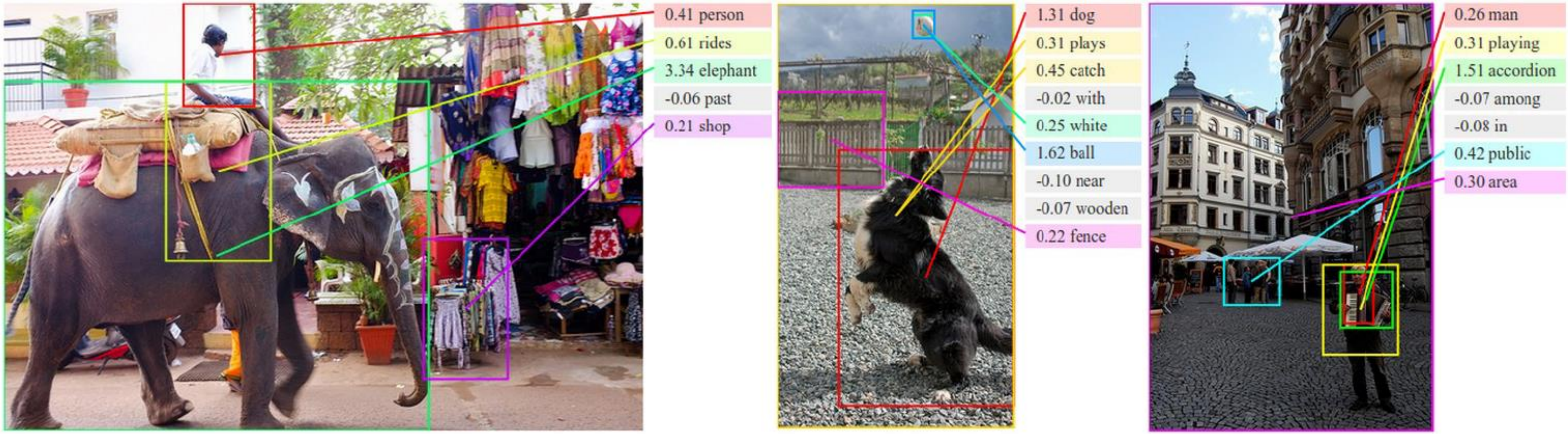

Figure 5. Example alignments predicted by our model. For every test image above, we retrieve the most compatible test sentence and visualize the highest-scoring region for each word (before MRF smoothing described in Section 3.1.4) and the associated scores ($v_i^T s_t$). We hide the alignments of low-scoring words to reduce clutter. We assign each region an arbitrary color.

Predicted image descriptions

Figure 6. Example sentences generated by the multimodal RNN for test images. We provide many more examples on our project page.

Predicted region descriptions

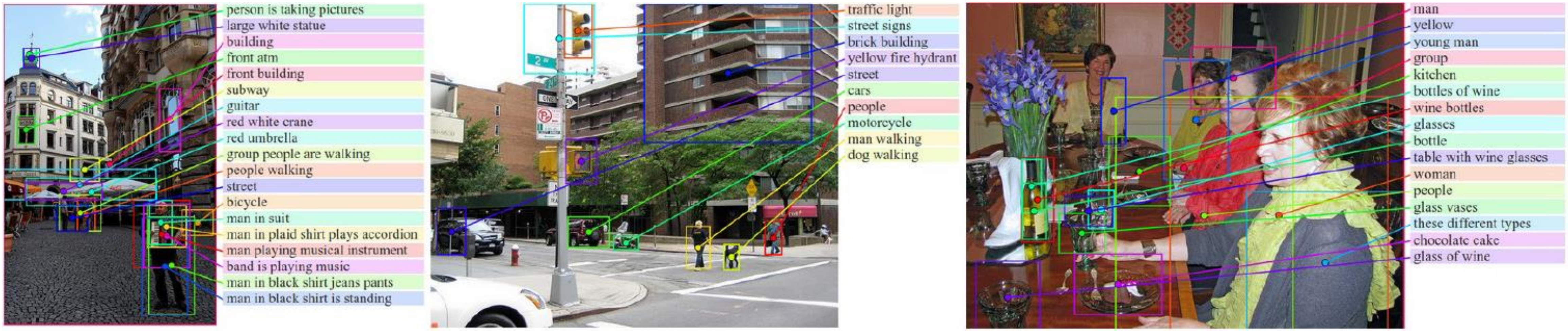

Figure 7. Example region predictions. We use our region-level multimodal RNN to generate text (shown on the right of each image) for some of the bounding boxes in each image. The lines are grounded to centers of bounding boxes and the colors are chosen arbitrarily.

Model Structure

How did they build it?

Shared Embedding Space

- Goal is to learn a single multimodal embedding space
  - Whole images and image regions
  - Word representations (including context)
- Model trained to align image region and word embeddings

Representing Images

- Pretrain R-CNN on ImageNet
  - Classification
- Fine tune on ImageNet Detection Challenge
  - Bounding boxes and labels
- Use top 19 detected locations and whole image
- Use 4096 dimension hidden activation just before the classifier

$$v = W_m[CNN_{\theta_c}(I_b)] + b_m, \quad (1)$$

Regions with CNN features (R-CNN)

- Girshick et al. (2014) Rich feature hierarchies for accurate object detection and semantic segmentation
  - https://arxiv.org/pdf/1311.2524.pdf
- Three modules
  - Generate region proposals (independent of category)
  - Generate fixed-length embeddings of variable sized regions
  - SVM
- Uijlings et al. (2012) Selective Search for Object Recognition
  - SIFT and HOG fed into SVM
  - Generates region proposals
  - https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf

Representing sentences

- Use bidirectional RNN (BRNN)
- Embeddings fixed using word2vec

$$x_t = W_w \mathbb{I}_t \tag{2}$$

$$e_t = f(W_e x_t + b_e) \tag{3}$$

$$h_t^f = f(e_t + W_f h_{t-1}^f + b_f) \tag{4}$$

$$h_t^b = f(e_t + W_b h_{t+1}^b + b_b) \tag{5}$$

$$s_t = f(W_d(h_t^f + h_t^b) + b_d). \tag{6}$$

Alignment Score

- Sentence-image pair should have high score if words have support in the image
- Previous model (Karpathy) utilizes dot product between region and word, summed over regions in image and words in sentence

$$S_{kl} = \sum_{t \in g_l} \sum_{i \in g_k} max(0, v_i^T s_t). \qquad (7)$$

- Current model is simplified: sum over words the max over regions

$$S_{kl} = \sum_{t \in g_l} max_{i \in g_k} v_i^T s_t. \qquad (8)$$

Alignment Objective

- Max-margin loss attempts to assign a low score to misaligned pairs

$$\mathcal{C}(\theta) = \sum_k \Big[\underbrace{\sum_l max(0, S_{kl} - S_{kk} + 1)}_{\text{rank images}} \tag{9}$$

$$+ \underbrace{\sum_l max(0, S_{lk} - S_{kk} + 1)}_{\text{rank sentences}} \Big].$$

Decoding text alignments

- Treat alignment as latent variables in an MRF
- Beta is hyperparameter controlling bias for single-word alignments or aligning the entire sentence
- Assign words to best regions, while trying to keep nearby words in the same region

$$E(\mathbf{a}) = \sum_{j=1\ldots N} \psi_j^U(a_j) + \sum_{j=1\ldots N-1} \psi_j^B(a_j, a_{j+1}) \quad (10)$$

$$\psi_j^U(a_j = t) = v_i^T s_t \quad (11)$$

$$\psi_j^B(a_j, a_{j+1}) = \beta \mathbb{1}[a_j = a_{j+1}]. \quad (12)$$

Generating descriptions

- RNN takes image pixels and generates a sequence of outputs
- Output probabilities over words (plus an END token)
- Image context only provided at first time step
- Trained to minimize NLL of target descriptions inferred by alignment model

$$b_v = W_{hi}[CNN_{\theta_c}(I)] \tag{13}$$

$$h_t = f(W_{hx}x_t + W_{hh}h_{t-1} + b_h + \mathbb{1}(t=1) \odot b_v) \tag{14}$$

$$y_t = softmax(W_{oh}h_t + b_o). \tag{15}$$

Training Details

- Preprocessing
  - Convert to lower-case and discard non-alphanumeric
  - Filter words occurring less than 5 times
- Alignment model
  - SGD with momentum
  - Dropout in all layers except recurrent layers
  - Clip gradients elementwise at 5 (important)
- Generative RNN
  - RMSprop

Recap: multi-step training

- Images
  - Train image classification
  - Train image detection
- Sentences
  - Train word2vec representations
- Train alignment model (including sentence BRNN)
- Solve MRF to generate alignments
- Train generative RNN

Experiments and Results

What did it do?

Experiments

- Datasets
  - Flikr8k
  - Flikr30K
  - MSCOCO
- Each dataset annotated with 5 region snippets
  - Collected using Amazon MT
  - For testing only

Alignment Recall Results

Model	Image Annotation				Image Search			
	R@1	**R@5**	**R@10**	**Med *r***	**R@1**	**R@5**	**R@10**	**Med *r***
Flickr30K								
SDT-RNN (Socher et al. [49])	9.6	29.8	41.1	16	8.9	29.8	41.1	16
Kiros et al. [25]	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Mao et al. [38]	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Donahue et al. [8]	17.5	40.3	50.8	9	-	-	-	-
DeFrag (Karpathy et al. [24])	14.2	37.7	51.3	10	10.2	30.8	44.2	14
Our implementation of DeFrag [24]	19.2	44.5	58.0	6.0	12.9	35.4	47.5	10.8
Our model: DepTree edges	20.0	46.6	59.4	5.4	15.0	36.5	48.2	10.4
Our model: BRNN	**22.2**	**48.2**	**61.4**	**4.8**	**15.2**	**37.7**	**50.5**	**9.2**
Vinyals et al. [54] (more powerful CNN)	23	-	63	5	17	-	57	8
MSCOCO								
Our model: 1K test images	38.4	69.9	80.5	1.0	27.4	60.2	74.8	3.0
Our model: 5K test images	16.5	39.2	52.0	9.0	10.7	29.6	42.2	14.0

Generation Results

- Use BLEU score to compare predicted text to actual annotations

	Flickr8K				Flickr30K				MSCOCO 2014					
Model	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	B-1	B-2	B-3	B-4	METEOR	CIDEr
Nearest Neighbor	—	—	—	—	—	—	—	—	48.0	28.1	16.6	10.0	15.7	38.3
Mao et al. [38]	58	28	23	—	55	24	20	—	—	—	—	—	—	—
Google NIC [54]	63	41	27	—	66.3	42.3	27.7	18.3	66.6	46.1	32.9	24.6	—	—
LRCN [8]	—	—	—	—	58.8	39.1	25.1	16.5	62.8	44.2	30.4	—	—	—
MS Research [12]	—	—	—	—	—	—	—	—	—	—	—	21.1	20.7	—
Chen and Zitnick [5]	—	—	—	14.1	—	—	—	12.6	—	—	—	19.0	20.4	—
Our model	57.9	38.3	24.5	16.0	57.3	36.9	24.0	15.7	62.5	45.0	32.1	23.0	19.5	66.0

Generated captions

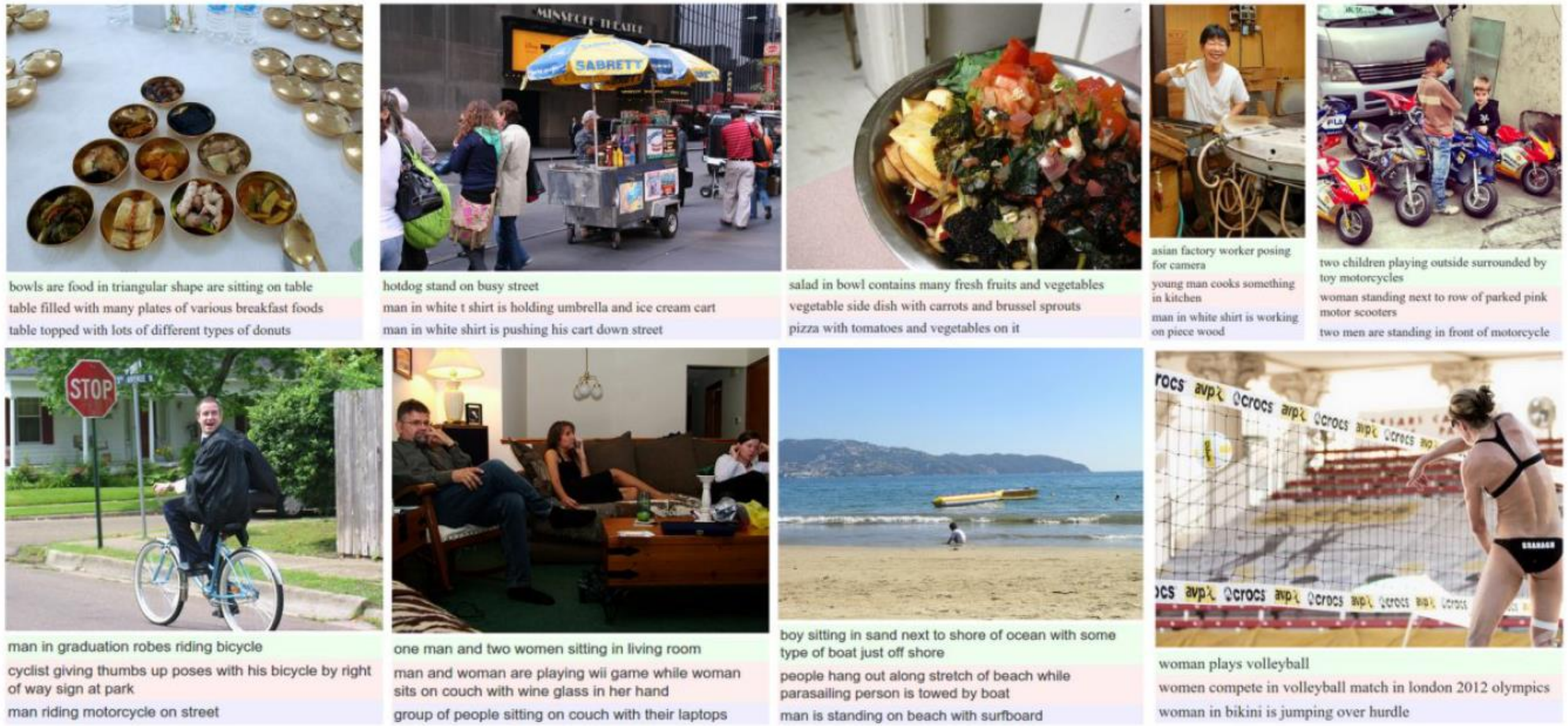

Figure 12. Additional examples of captions on the level of full images. Green: Human ground truth. Red: Top-scoring sentence from training set. Blue: Generated sentence.

Query by snippets

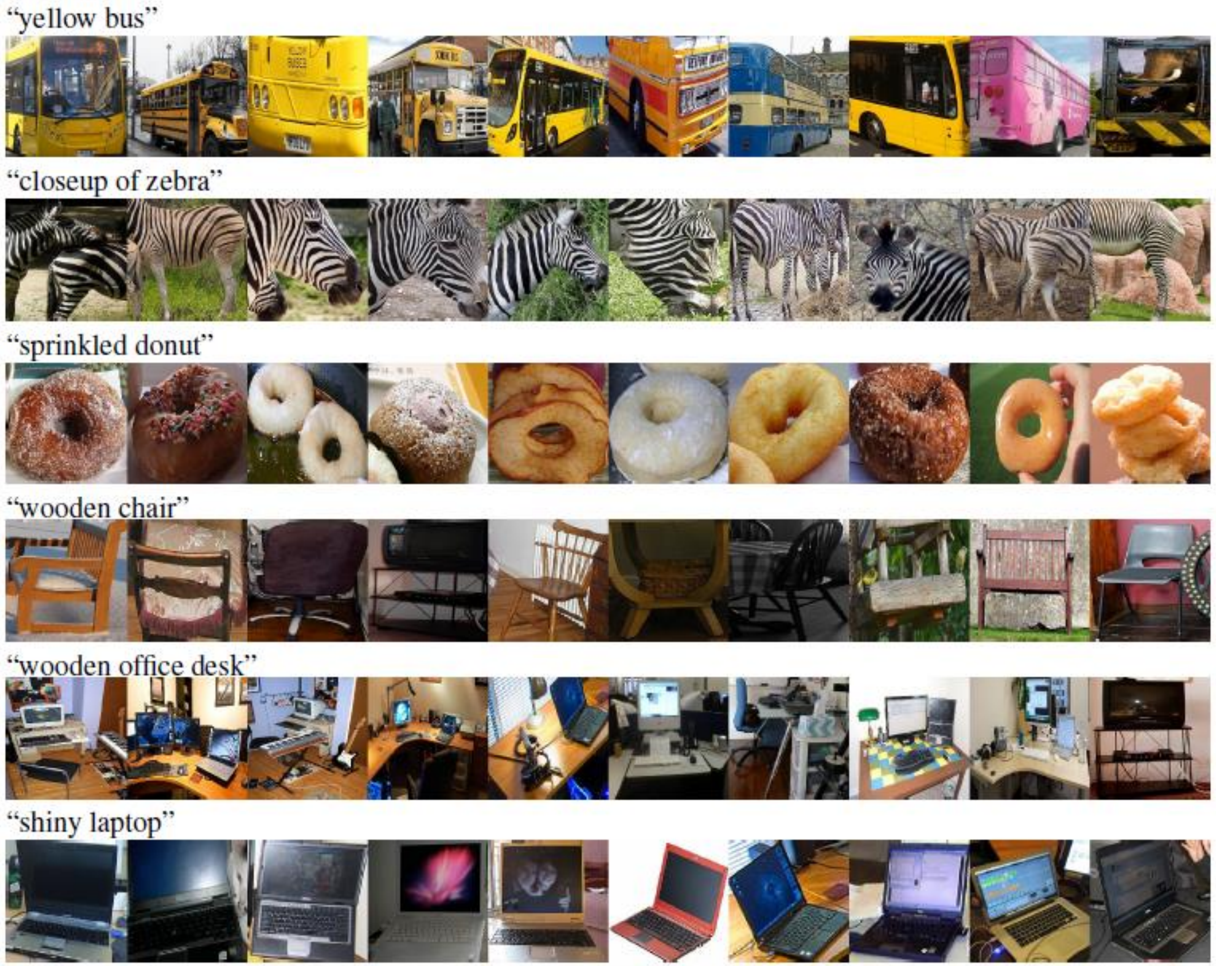

Figure 9. Examples of highest scoring regions for queried snippets of text, on 5,000 images of our MSCOCO test set.

Inferred Alignments

Figure 11. Additional examples of alignments. For each query test image above we retrieve the most compatible sentence from the test set and show the alignments.

Generated Region Captions

Figure 13. Additional examples of region captions on the test set of Flickr30K.

Discussion

What does this mean?

Discussion

- Simplified cost function improves performance
- BRNN outperforms dependency tree relations
- Embeddings of important words (“kayaking”,”pumpkins”) are larger magnitude than stop words (“now”,”but”,”simply”)

Comments

- Region annotations are independently generated
  - How would you model dependent annotations?
- Lower levels are not retrained while training higher levels
  - Should alignment decisions affect what regions to label?
  - Is it possible to train something this complicated as a full stack?
- Training data includes bounding boxes
  - How could you infer bounding boxes if they were not provided?
- Limiting the embeddings using weight clipping seems dangerous
  - If you need to prevent large embeddings, there are many options

Questions?

Don’t be shy