Semantic Compositional Networks for Visual Captioning

Zhe Gan

Advisor: Dr. Lawrence Carin

Ph.D. Preliminary Exam

April 7th, 2017

Outline

1 Introduction

2 Proposed model

3 Experiments

4 Conclusion

Problem of interest

- Can we build a model that is able to generate a natural sentence description of an input image/video?
- Intersection between CV and NLP
- Retrieval-based and template-based methods: *cannot* generate novel captions

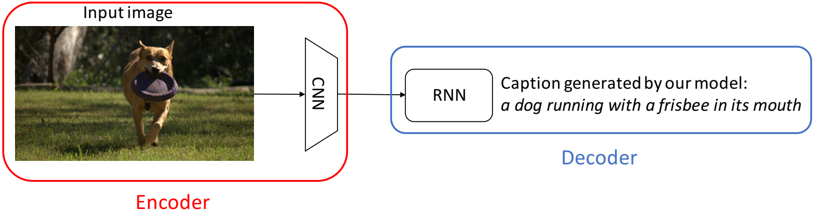
Input image

Human captions:

1. a very cute brown dog with a disc in its mouth
2. a dog running in the grass with a frisbee in his mouth
3. a dog carrying a frisbee in its mouth running on a grass lawn
4. a dog in a grassy field carrying a frisbee
5. a brown dog walking across a green field with a frisbee in its mouth

Problem of interest

- Neural-network-based method: the encoder-decoder framework [10, 12] (by Google)
- Follow-up work: Standford [5], Berkeley [1], UCLA&Baidu [7], Montreal&Toronto [14], MSR [2], etc.



Review of RNN for image captioning

- Consider an image $\mathbf{I}$, with associated caption $\mathbf{X}$.
- Image $\mathbf{I}$ is often represented by a feature vector $\boldsymbol{v}(\mathbf{I})$, obtained by a pretrained CNN.
- $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)$, with $\boldsymbol{x}_t$ a 1-of-V ("one-hot") encoding vector.
- $\boldsymbol{x}_t$ is linearly embedded into an n_x-dimensional real-valued vector $\boldsymbol{w}_t = \mathbf{W}_e \boldsymbol{x}_t$, where $\mathbf{W}_e \in \mathbb{R}^{n_x \times V}$ is a word embedding matrix (learned).

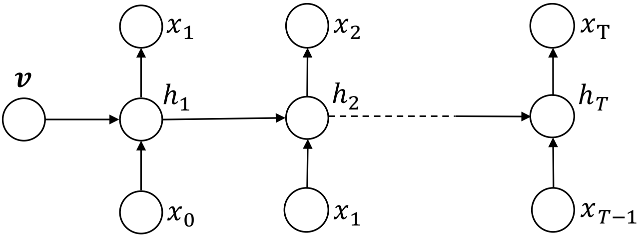
Review of RNN for image captioning

- The probability of caption $\mathbf{X}$ given image feature vector $\boldsymbol{v}$ is

$$p(\mathbf{X}|\mathbf{I}) = \prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{t-1}, \boldsymbol{v}) \,, \qquad (1)$$

- Each conditional $p(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{v})$ is specified as softmax$(\mathbf{V}\boldsymbol{h}_t)$

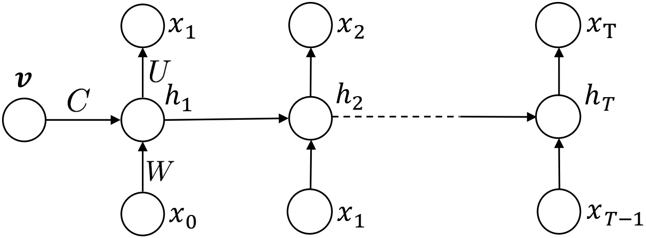
$$\boldsymbol{h}_t = \mathcal{H}(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{v}) \qquad (2)$$



Review of RNN for image captioning

- $\boldsymbol{x}_0$ is defined as a special start-of-the-sentence token
- We also define a special end-of-the-sentence token
- Consider an RNN with a simple transition function $\mathcal{H}(\cdot)$

$$\boldsymbol{h}_t = \sigma(\mathbf{W}\boldsymbol{x}_{t-1} + \mathbf{U}\boldsymbol{h}_{t-1} + I(t=1)\cdot\mathbf{C}\boldsymbol{v}), \qquad (3)$$



Outline

1 Introduction

2 Proposed model

3 Experiments

4 Conclusion

Proposed model

1. First do image tagging, then image captioning
   - Similiar ideas also used in [2] (by MSR), [13, 15].
2. How to integrate detected semantic concepts into the caption generation process
   - Our key contribution

Semantic concept detection

- First select a set of tags from the captions in the training set
  - Nouns: snow, man, dog, room, ocean etc.
  - Verbs: skiing, riding, brushing, holding, running etc.
  - Adjectives: white, cute, young, large, wooden etc.
- We treat image tagging as a multi-label classification task
- Let $\boldsymbol{y}_i = [y_{i1}, \ldots, y_{iK}] \in \{0,1\}^K$ be the label vector
  - $y_{ik} = 1$ if the image is annotated with tag k
  - $y_{ik} = 0$ otherwise.
- Let $\boldsymbol{v}_i$ represent the image feature vector

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K}\Big(y_{ik}\log s_{ik} + (1-y_{ik})\log(1-s_{ik})\Big)\,, \tag{4}$$

- $\boldsymbol{s}_i = \sigma(f(\boldsymbol{v}_i))$ is the semantic feature vector.

Semantic concept detection: Examples

outdoor (0.998) mountain (0.973)
person (0.93) man (0.829)
grass (0.813) red (0.543)
carrying (0.404) dirt (0.403)
holding (0.356) riding (0.297)

table (0.996) pizza (0.996)
food (0.989) indoor (0.976)
sitting (0.926) wooden (0.655)
slice (0.527) piece (0.506)

Semantic compositional network

- Basic RNN:

$$\boldsymbol{h}_t = \sigma(\mathbf{W}\boldsymbol{x}_{t-1} + \mathbf{U}\boldsymbol{h}_{t-1} + I(t=1)\cdot\mathbf{C}\boldsymbol{v}), \tag{5}$$

- How to assemble the meanings of individual tags to generate the caption?
- Simple solution: Feed the tags as an *initialization* step into the RNN decoder [13]

$$\boldsymbol{h}_t = \sigma(\mathbf{W}\boldsymbol{x}_{t-1} + \mathbf{U}\boldsymbol{h}_{t-1} + I(t=1)\cdot(\mathbf{C}_1\boldsymbol{v} + \mathbf{C}_2\boldsymbol{s})), \tag{6}$$

- Better approach: Semantic Compositional Network (SCN)

$$\boldsymbol{h}_t = \sigma(\mathbf{W}(\boldsymbol{s})\boldsymbol{x}_{t-1} + \mathbf{U}(\boldsymbol{s})\boldsymbol{h}_{t-1} + I(t=1)\cdot\mathbf{C}\boldsymbol{v}). \tag{7}$$

Semantic compositional network

- Semantic compositional network

$$\boldsymbol{h}_t = \sigma(\mathbf{W}(\boldsymbol{s})\boldsymbol{x}_{t-1} + \mathbf{U}(\boldsymbol{s})\boldsymbol{h}_{t-1} + I(t=1)\cdot \mathbf{C}\boldsymbol{v})\,, \qquad (8)$$

- Making $\mathbf{W}(\boldsymbol{s})$ and $\mathbf{U}(\boldsymbol{s})$ *adaptive* to the input image
- Training a *personalized* RNN for each input image
- How to design $\mathbf{W}(\boldsymbol{s})$ and $\mathbf{U}(\boldsymbol{s})$?
    - $\mathbf{W}(\boldsymbol{s})$ and $\mathbf{U}(\boldsymbol{s})$ are ensembles of tag-dependent weight matrices, subjective to the probabilities that the tags are present in the image, according to the semantic-concept vector $\boldsymbol{s}$.

Semantic compositional network

- Given $\boldsymbol{s} \in \mathbb{R}^K$, we define two weight tensors $\mathbf{W}_\mathcal{T} \in \mathbb{R}^{n_h \times n_x \times K}$ and $\mathbf{U}_\mathcal{T} \in \mathbb{R}^{n_h \times n_h \times K}$.
- $\mathbf{W}(\boldsymbol{s}) \in \mathbb{R}^{n_h \times n_x}$ and $\mathbf{U}(\boldsymbol{s}) \in \mathbb{R}^{n_h \times n_h}$ can be specified as

$$\mathbf{W}(\boldsymbol{s}) = \sum_{k=1}^{K} s_k \mathbf{W}_\mathcal{T}[k], \; \mathbf{U}(\boldsymbol{s}) = \sum_{k=1}^{K} s_k \mathbf{U}_\mathcal{T}[k] \,, \qquad (9)$$

- Can be interpreted as *jointly* training an ensemble of K RNNs in total.
- Though appealing, the number of parameters is proportional to K, which is prohibitive for large K (*e.g.*, $K = 1000$ for COCO).

Semantic compositional network

- We adopt ideas from [8] to factorize $\mathbf{W}(\boldsymbol{s})$ and $\mathbf{U}(\boldsymbol{s})$ as

$$\mathbf{W}(\boldsymbol{s}) = \mathbf{W}_a \cdot \mathrm{diag}(\mathbf{W}_b \boldsymbol{s}) \cdot \mathbf{W}_c \,, \tag{10}$$

$$\mathbf{U}(\boldsymbol{s}) = \mathbf{U}_a \cdot \mathrm{diag}(\mathbf{U}_b \boldsymbol{s}) \cdot \mathbf{U}_c \,, \tag{11}$$

  $\mathbf{W}_a \in \mathbb{R}^{n_h \times n_f}$, $\mathbf{W}_b \in \mathbb{R}^{n_f \times K}$ and $\mathbf{W}_c \in \mathbb{R}^{n_f \times n_x}$. Similiarly, $\mathbf{U}_a \in \mathbb{R}^{n_h \times n_f}$, $\mathbf{U}_b \in \mathbb{R}^{n_f \times K}$ and $\mathbf{U}_c \in \mathbb{R}^{n_f \times n_h}$.
- $\mathbf{W}_a$ and $\mathbf{W}_c$ are shared among all the captions, effectively capturing common linguistic patterns
- $\mathrm{diag}(\mathbf{W}_b \boldsymbol{s})$, accounts for semantic aspects of the image under test, captured by $\boldsymbol{s}$
- The RNN weight matrices that correspond to each semantic concept share "structure"

Semantic compositional network: SCN-RNN

- Let $\boldsymbol{w}_{bk}$ represent the kth column of $\mathbf{W}_b$, then

$$\mathbf{W}(\boldsymbol{s}) = \sum_{k=1}^{K} s_k \mathbf{W}_{\mathcal{T}}[k], \tag{12}$$

$$\mathbf{W}(\boldsymbol{s}) = \sum_{k=1}^{K} s_k [\mathbf{W}_a \cdot \mathrm{diag}(\boldsymbol{w}_{bk}) \cdot \mathbf{W}_c] \, . \tag{13}$$

- In terms of implementation, we introduce *multiplicative* connections

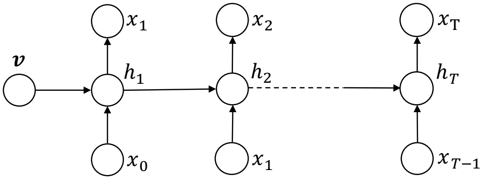
$$\tilde{\boldsymbol{x}}_{t-1} = \mathbf{W}_b \boldsymbol{s} \odot \mathbf{W}_c \boldsymbol{x}_{t-1} \, , \tag{14}$$

$$\tilde{\boldsymbol{h}}_{t-1} = \mathbf{U}_b \boldsymbol{s} \odot \mathbf{U}_c \boldsymbol{h}_{t-1} \, , \tag{15}$$

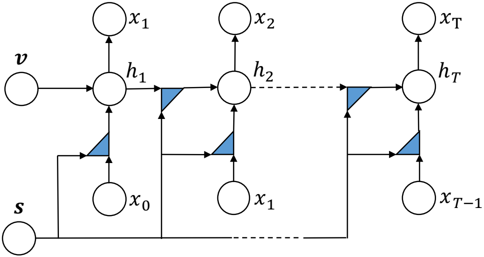
$$\boldsymbol{z} = I(t=1) \cdot \mathbf{C} \boldsymbol{v} \, , \tag{16}$$

$$\boldsymbol{h}_t = \sigma(\mathbf{W}_a \tilde{\boldsymbol{x}}_{t-1} + \mathbf{U}_a \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{z}) \, . \tag{17}$$

Semantic compositional network: Comparsion



(a) Basic RNN



(b) SCN-RNN

Semantic compositional network: SCN-LSTM

- Computational complexity
  - The number of parameters in the basic RNN model is $n_h \cdot (n_x + n_h)$
  - The number of parameters in the SCN-RNN model is $n_f \cdot (n_x + 2K + 3n_h)$
  - In experiments, we set $n_f = n_h$. Therefore, the additional number of parameters is $2 \cdot n_h \cdot (n_h + K)$
- Remind that we are using simple RNN transition functions

$$\boldsymbol{h}_t = \sigma(\mathbf{W}_a \tilde{\boldsymbol{x}}_{t-1} + \mathbf{U}_a \tilde{\boldsymbol{h}}_{t-1} + \boldsymbol{z}) \tag{18}$$

- In order to capture long-term dependencies, we introduce Long Short-Term Memory (LSTM) [4] units and generalize SCN-RNN to SCN-LSTM.

LSTM

- How to design $\boldsymbol{h}_t = \mathcal{H}(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1})$?
- Long Short-Term Memory (LSTM) [4]:
  - Learn to remember and forget adaptively

$$\boldsymbol{i}_t = \sigma(\mathbf{W}_i \boldsymbol{x}_{t-1} + \mathbf{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i)\,, \tag{19}$$

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_f \boldsymbol{x}_{t-1} + \mathbf{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f)\,, \tag{20}$$

$$\boldsymbol{o}_t = \sigma(\mathbf{W}_o \boldsymbol{x}_{t-1} + \mathbf{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o)\,, \tag{21}$$

$$\tilde{\boldsymbol{c}}_t = \tanh(\mathbf{W}_c \boldsymbol{x}_{t-1} + \mathbf{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c)\,, \tag{22}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t\,, \tag{23}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)\,. \tag{24}$$

Semantic compositional network: SCN-LSTM

We define $\boldsymbol{h}_t = \mathcal{H}(\boldsymbol{x}_{t-1}, \boldsymbol{h}_{t-1}, \boldsymbol{v}, \boldsymbol{s})$ as

$$\boldsymbol{i}_t = \sigma(\mathbf{W}_{ia}\tilde{\boldsymbol{x}}_{i,t-1} + \mathbf{U}_{ia}\tilde{\boldsymbol{h}}_{i,t-1} + \boldsymbol{z})\,, \tag{25}$$

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_{fa}\tilde{\boldsymbol{x}}_{f,t-1} + \mathbf{U}_{fa}\tilde{\boldsymbol{h}}_{f,t-1} + \boldsymbol{z})\,, \tag{26}$$

$$\boldsymbol{o}_t = \sigma(\mathbf{W}_{oa}\tilde{\boldsymbol{x}}_{o,t-1} + \mathbf{U}_{oa}\tilde{\boldsymbol{h}}_{o,t-1} + \boldsymbol{z})\,, \tag{27}$$

$$\tilde{\boldsymbol{c}}_t = \sigma(\mathbf{W}_{ca}\tilde{\boldsymbol{x}}_{c,t-1} + \mathbf{U}_{ca}\tilde{\boldsymbol{h}}_{c,t-1} + \boldsymbol{z})\,, \tag{28}$$

$$\boldsymbol{c}_t = \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t + \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1}\,, \tag{29}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)\,, \tag{30}$$

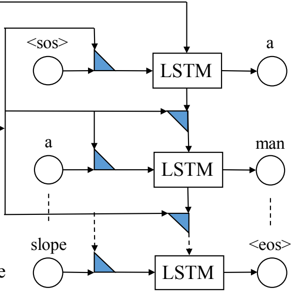
where $\boldsymbol{z} = I(t=1)\cdot\mathbf{C}\boldsymbol{v}$. For $\star = i, f, o, c$, we define

$$\tilde{\boldsymbol{x}}_{\star,t-1} = \mathbf{W}_{\star b}\boldsymbol{s} \odot \mathbf{W}_{\star c}\boldsymbol{x}_{t-1}\,, \tag{31}$$

$$\tilde{\boldsymbol{h}}_{\star,t-1} = \mathbf{U}_{\star b}\boldsymbol{s} \odot \mathbf{U}_{\star c}\boldsymbol{h}_{t-1}\,. \tag{32}$$

Semantic compositional network: Illustration

snow	1.000
skiing	0.993
man	0.917
slope	0.898
person	0.889
hill	0.808
covered	0.750
riding	0.627



Generated caption: a man riding skis down a snow covered slope

Semantic Composition

Detected semantic concepts:
person (0.998), baby (0.983), holding (0.952), small (0.697), sitting (0.638), toothbrush (0.538), child (0.502), mouth (0.438)

Semantic composition:
1. Only using "**baby**": *a baby in a*
2. Only using "**holding**": *a person holding a hand*
3. Only using "**toothbrush**": *a pair of toothbrush*
4. Only using "**mouth**": *a man with a toothbrush*
5. Using "**baby**" and "**mouth**": *a baby brushing its teeth*

Overall caption generated by the SCN:
a baby holding a toothbrush in its mouth

Influence the caption by changing the tag:
6. Replace "**baby**" with "**girl**": *a little girl holding a toothbrush in her mouth*
7. Replace "**toothbrush**" with "**baseball**": *a baby holding a baseball bat in his hand*
8. Replace "**toothbrush**" with "**pizza**": *a baby holding a piece of pizza in his mouth*

Extension to video captioning

- We use a two-dimensional (2D) *and* a three-dimensional (3D) CNN to extract visual features of video frames/clips
- We then perform a mean pooling process over all 2D CNN features and 3D CNN features, to generate two feature vectors (one from 2D CNN features and the other from 3D CNN features)
- The representation of each video is produced by concatenating these two features

Outline

1. Introduction
2. Proposed model
3. Experiments
4. Conclusion

Datasets

- COCO: 120K images
  - Each image is annotated with at least 5 captions.
  - Vocabulary size: 8791
  - Testing: 40K blind test
  - Training:
    - Official recommendation: 80K training, 40K development
    - Our setup: 110K training, 5K dev-validation, 5K dev-test
- Flickr30k: 30K images
  - 1000 for validation, 1000 for test, the rest for training
  - Vocabulary size: 7414
- Youtube2Text: 1970 Youtube clips
  - 1200 for training, 100 for validation, 670 for test
  - Vocabulary size: 12594

Setup

- For image representation, we use ResNet-152 [3], pretrained on the ImageNet dataset [9].
- For video representation, we also utilize a 3D CNN (C3D) [11], pretrained on Sports-1M video dataset [6].
- In testing, we use beam search for caption generation, and set the beam size to $k = 5$.

Quantitative results

Our SCN model achieves the state-of-the-art results.

Methods	COCO					
	B-1	B-2	B-3	B-4	M	C
NIC [48]	0.666	0.451	0.304	0.203	–	–
m-RNN [29]	0.67	0.49	0.35	0.25	–	–
Hard-Attention [52]	0.718	0.504	0.357	0.250	0.230	–
ATT [54]	0.709	0.537	0.402	0.304	0.243	–
Att-CNN+LSTM [49]	0.74	0.56	0.42	0.31	0.26	0.94
LSTM-R	0.698	0.525	0.390	0.292	0.238	0.889
LSTM-T	0.716	0.546	0.411	0.312	0.250	0.952
LSTM-RT	0.724	0.555	0.419	0.316	0.252	0.970
LSTM-RT_2	0.730	0.568	0.430	0.322	0.249	0.977
SCN-LSTM	0.728	0.566	0.433	0.330	0.257	1.012
SCN-LSTM Ensemble of 5	**0.741**	**0.578**	**0.444**	**0.341**	**0.261**	**1.041**

Quantitative results

Model	BLEU-1		BLEU-2		BLEU-3		BLEU-4		METEOR		ROUGE-L		CIDEr-D	
	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40	c5	c40
SCN-LSTM	**0.740**	**0.917**	**0.575**	**0.839**	**0.436**	**0.739**	**0.331**	**0.631**	**0.257**	**0.348**	**0.543**	**0.696**	**1.003**	**1.013**
ATT	0.731	0.900	0.565	0.815	0.424	0.709	0.316	0.599	0.250	0.335	0.535	0.682	0.943	0.958
OV	0.713	0.895	0.542	0.802	0.407	0.694	0.309	0.587	0.254	0.346	0.530	0.682	0.943	0.946
MSR Cap	0.715	0.907	0.543	0.819	0.407	0.710	0.308	0.601	0.248	0.339	0.526	0.680	0.931	0.937

Table 2: Comparison to published state-of-the-art image captioning models on the blind test set as reported by the COCO test server. SCN-LSTM is our model. ATT refers to ATT VC [54], OV refers to OriolVinyals [48], and MSR Cap refers to MSR Captivator [9].

Model	B-4	M	C
S2VT [46]	–	0.292	–
LSTM-E [32]	0.453	0.310	–
GRU-RCN [3]	0.479	0.311	0.678
h-RNN [56]	0.499	0.326	0.658
LSTM-R	0.448	0.310	0.640
LSTM-C	0.445	0.309	0.644
LSTM-CR	0.469	0.317	0.688
LSTM-T	0.473	0.324	0.699
LSTM-CRT	0.475	0.316	0.647
LSTM-CRT$_2$	0.469	0.326	0.706
SCN-LSTM	0.502	0.334	0.770
SCN-LSTM Ensemble of 5	**0.511**	**0.335**	**0.777**

Table 3: Results on BLEU-4 (B-4), METEOR (M) and CIDEr-D (C) metrices compared to other state-of-the-art results and baselines on Youtube2Text.

Qualitative analysis

SCN can adjust the caption smoothly as the tags are modified.

	**Tags:** dog (1), grass (0.996), laying (0.97), outdoor (0.943), next (0.788), sitting (0.651), lying (0.542), white (0.507)		**Tags:** road (1), decker (1), double (0.999), bus (0.996), red (0.996), street (0.926), building (0.859), driving (0.796)

Caption generated by our model: *a dog laying on the ground next to a frisbee* **Semantic composition:** 1. Replace "dog" with "**cat**": *a white cat laying on the ground* 2. Replace "grass" with "**bed**": *a white dog laying on top of a bed* 3. Replace "grass" with "**laptop**": *a dog laying on the ground next to a laptop*	**Caption generated by our model:** *a red double decker bus driving down a street* **Semantic composition:** 1. Replace "red" with "**blue**": *a blue double decker bus driving down a street* 2. Replace "bus" with "**train**": *a red train traveling down a city street* 3. Replace "road" and "street" with "**ocean**": *a red bus is driving in the ocean*

Qualitative analysis

Importance of using detected tags

Tags:
book (1), shelf (1), table (0.965), sitting (0.955), person (0.955), library (0.908), room (0.829), front (0.464)

Generated captions:
LSTM-R: a young girl is playing a video game
LSTM-RT: a group of people sitting at a table
SCN-LSTM: two women sitting at a table in a library

Tags:
grass (1), red (0.982), fire (0.953), hydrant (0.852), dog (0.723), standing (0.598), next (0.476), field (0.341)

Generated captions:
LSTM-R: a dog that is sitting on the ground
LSTM-RT$_2$: a dog standing next to a fire hydrant
SCN-LSTM: a dog standing next to a red fire hydrant

Qualitative analysis

Importance of using visual features

Tags:
indoor (0.952), dog (0.828), sitting (0.647), stuffed (0.602), white (0.544), next (0.527), laying (0.509), cat (0.402)

Generated captions:
SCN-LSTM-T: a dog laying on top of a stuffed animal
SCN-LSTM: a teddy bear laying on top of a stuffed animal

Tags:
snow(1), outdoor (0.992), covered (0.847), nature (0.812), skiing (0.61), man (0.451), pile (0.421), building (0.369)

Generated captions:
SCN-LSTM-T: a person that is standing in the snow
SCN-LSTM: a stop sign is covered in the snow

Video captioning

a man is playing with a dog

the men are playing soccer

a girl is playing a guitar

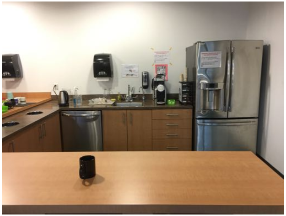
a man is pushing a car

Image captioning in the wild

A tall tower with a clock on it

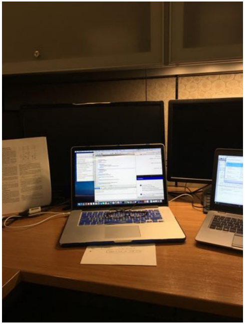
A group of people playing a game of basketball

A plate of food on a table



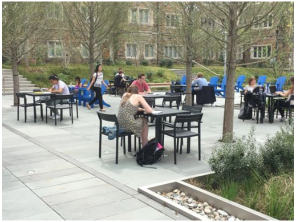
A kitchen with a sink and a refrigerator

Image captioning in the wild



A laptop computer sitting on top of a wooden desk

A statue of a horse in a field



A group of people sitting on a park bench

A red stop sign sitting on the side of a road

Outline

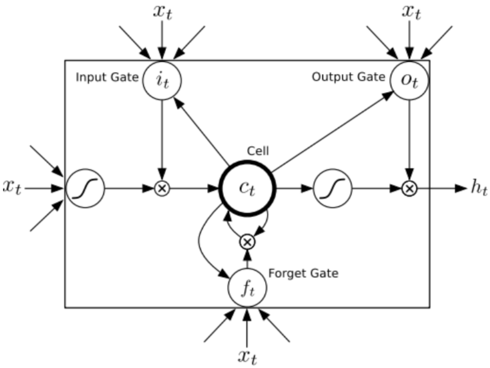
1. Introduction
2. Proposed model
3. Experiments
4. Conclusion

Summary and future work

- Summary
  - We propose SCN, which extends each weight matrix of the conventional LSTM to be a three-way matrix product, with one of these matrices dependent on the inferred tags.
  - SCN can be viewed an ensemble of tag-dependent LSTM bases
  - We achieve state-of-the-art results
- Future work
  - Using adversarial loss (GAN) instead of cross-entropy loss (MLE)
  - Joint image captioning and text to image synthesis

Backup: LSTM

- Input gate: scales input to cell (write)
- Output gate: scales output from cell (read)
- Forget gate: scales old cell value (reset)



References I

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell.
Long-term recurrent convolutional networks for visual recognition and description.
In *CVPR*, 2015.

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al.
From captions to visual concepts and back.
In *CVPR*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition.
In *CVPR*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber.
Long short-term memory.
Neural computation, 1997.

Andrej Karpathy and Li Fei-Fei.
Deep visual-semantic alignments for generating image descriptions.
In *CVPR*, 2015.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei.
Large-scale video classification with convolutional neural networks.
In *CVPR*, 2014.

References II

Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille.
Deep captioning with multimodal recurrent neural networks (m-rnn).
In *ICLR*, 2015.

Roland Memisevic and Geoffrey Hinton.
Unsupervised learning of image transformations.
In *CVPR*, 2007.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al.
Imagenet large scale visual recognition challenge.
IJCV, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le.
Sequence to sequence learning with neural networks.
In *NIPS*, 2014.

D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri.
Learning spatiotemporal features with 3d convolutional networks.
In *ICCV*, 2015.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan.
Show and tell: A neural image caption generator.
In *CVPR*, 2015.

Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel.
What value do explicit high level concepts have in vision to language problems?
In *CVPR*, 2016.

References III

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio.
Show, attend and tell: Neural image caption generation with visual attention.
In *ICML*, 2015.

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo.
Image captioning with semantic attention.
In *CVPR*, 2016.