Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models

Jamie Ryan Kiros, Ruslan Salakhutdinov, Richard Zemel

Presentation by David Madras

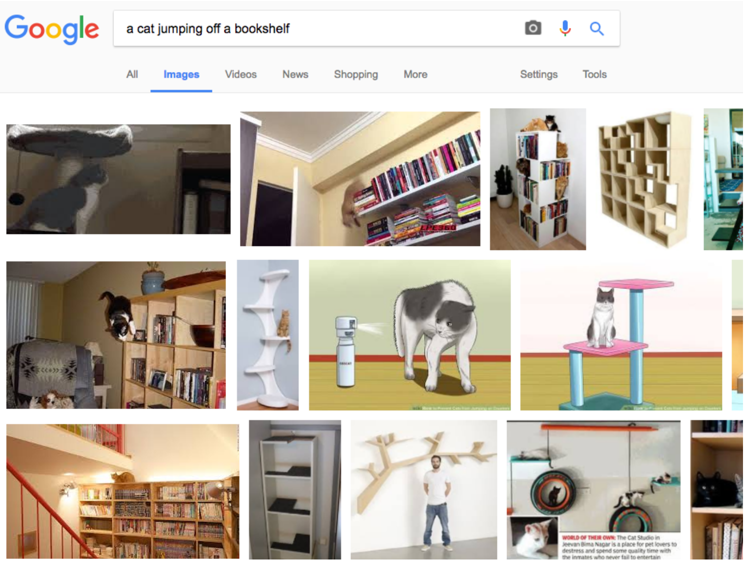
University of Toronto

January 25, 2017

Image Captioning

???????

Image Retrieval



Introduction: Captioning and Retrieval

- **Image captioning**: the challenge of generating descriptive sentences for images
- Must consider spatial relationships between objects
- Also should generate grammatical, sensible phrases
- **Image retrieval** is related: given a query sentence, find the most relevant pictures in a database

Figure 1: Caption Example: A cat jumping off a bookshelf

Approaches to Captioning

1. Template based methods
   - Begin with several pre-determined sentence templates
   - Fill these in with object detection, analyzing spatial relationships
   - Less generalizable, captions don't feel very fluid, "human"
2. Composition-based methods
   - Extract and re-compose components of relevant, existing captions
   - Try to find the most "expressive" components
   - e.g. TREETALK [Kuznetsova et al., 2014] - uses tree fragments
3. Neural Network Methods
   - Sample from a conditional neural language model
   - Generate description sentence by conditioning on the image

The paper we'll talk about today fits (unsurprisingly) into the Neural Network Methods category.

High-Level Approach

- Kiros et al. take approach inspired by translation: images and text are different "languages" that can express the same concept
- Sentences and images are embedded in same representation space; similar underlying concepts should have similar representations
- To caption an image:
  1. Find that image's embedding
  2. Sample a point near that embedding
  3. Generate text from that point
- To do image retrieval for a sentence:
  1. Find that sentence's embedding
  2. Do a nearest neighbour search in the embedding space for images in our database

Encoder-Decoder Model

- An encoder-decoder model has two components
- **Encoder functions** which transform data into a representation space
- **Decoder functions** which transform a vector from representation space into data

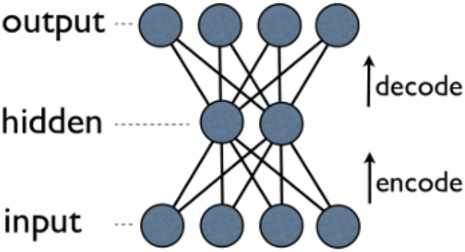


Figure 2: The basic encoder-decoder structure

Encoder-Decoder Model

- Kiros et al. learn these functions using neural networks. Specifically:
  - **Encoder for sentences**: recurrent neural network (RNN) with long short-term memory (LSTM)
  - **Encoder for images**: convolutional neural network (CNN)
  - **Decoder for sentences**: Structure-Content Neural Language Model
  - No decoder for images in this model - that's a separate question

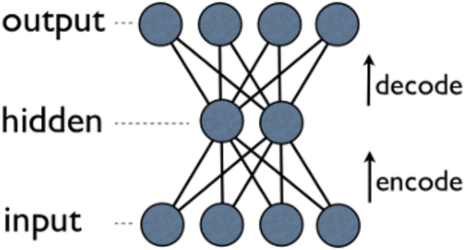


Figure 3: The basic encoder-decoder structure

Obligatory Model Architecture Slide

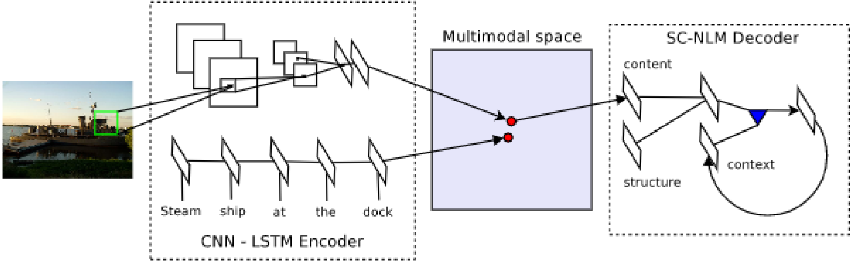


Figure 4: The model for captioning/retrieval proposed by Kiros et al.

Recurrent Neural Networks (RNNs)

- Recurrent neural networks have loops in them
- We propogate information between time steps
- Allows us to use neural networks on **sequential, variable-length** data
- Our current state is influenced by input *and* all past states

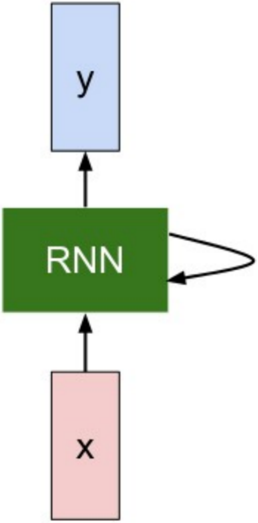


Figure 5: A basic (vanilla) RNN

Image from Andrej Karpathy

Recurrent Neural Networks (RNNs)

- By unrolling the network through time, an RNN has similar structure to a feedforward NN
- Weights are shared throughout time - can lead to vanishing/exploding gradient problem
- RNN's are Turing-complete - can simulate arbitrary programs (...in theory)

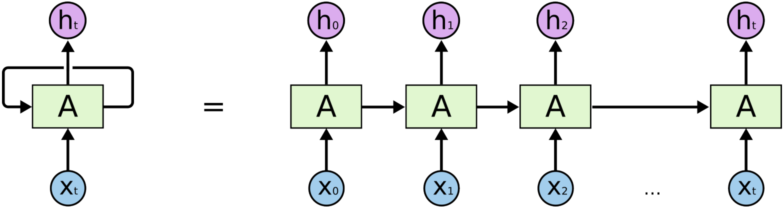


Figure 6: RNN unrolled through time

Image from Chris Olah

RNNs for Language Models

- Language is a natural application for RNNs, as it takes a sequential, variable-length form

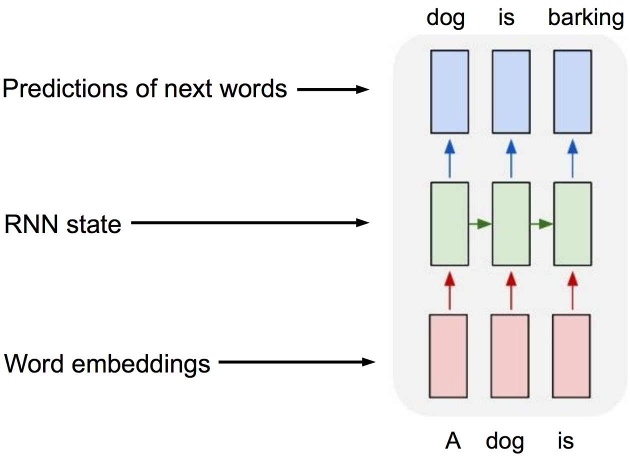


Image from Jamie Kiros

RNNs for Conditional Language Models

- We can condition our sentences on an alternate input

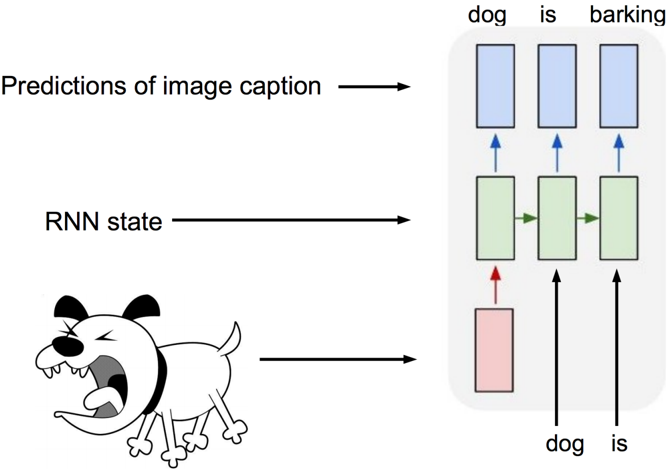


Image from Jamie Kiros

RNNs for Language Models: Encoders

- We can use RNNs to encode sentences in a high-dimensional representation space

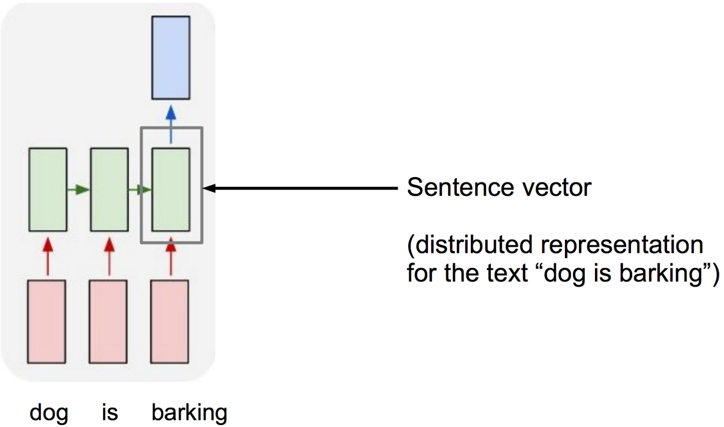
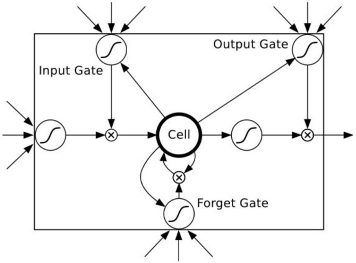


Image from Jamie Kiros

Long Short-Term Memory (LSTM)



Input gate: scales input to cell (write)

Output gate: scales output from cell (read)

Forget gate: scales old cell value (reset)

- Learning long-term dependencies with RNNs can be difficult
- LSTM cells [Hochreiter, 1997] can do a better job at this
- The network explicitly learns how much to "remember" or "forget" at each time step
- LSTMs also help with the vanishing gradient problem

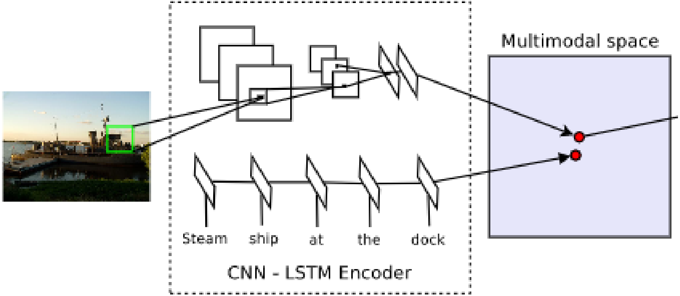
Image from Alex Graves

Learning Multimodal Distributed Representations

- Jointly optimize text/image encoders for images x, captions v
- $s(x, v)$ is cosine similarity, and v_k are a set of random captions which do **not** describe image x

$$\min_{\theta} \sum_{x,k} max(0, \alpha - s(x, v) + s(x, v_k)) + \sum_{v,k} max(0, \alpha - s(v, x) + s(v, x_k))$$

- Maximize similarity between x's embedding and its descriptions', and minimize similarity to all other sentences



Neural Language Decoders

- That's the encoding half of the model - any questions?
- Now we'll talk about the decoding half
- The authors describe two types of models: log-bilinear and multiplicative
- The model they ultimately use is based on the more complex multiplicative model, but I think it's helpful to explain both

Log-bilinear neural language models

- In sentence generation, we model the probability of the next word given the previous words - $P(w_n|w_{1:n-1})$
- We can represent each word as a K-dimensional vector w_i
- In an LBL, we make a linear prediction of w_n with

$$\hat{r} = \sum_{i=1}^{n-1} C_i w_i$$

  where $\hat{r}$ is the predicted representation of w_n, and C_i are context parameter matrices for each index
- We then use a softmax over all word representations r_i to get a probability distribution over the vocabulary

$$P(w_n = i|w_{1:n-1}) = \frac{\exp(\hat{r}^T w_i + b_i)}{\sum_j^V \exp(\hat{r}^T w_j + b_j)}$$

- We learn C_i through gradient descent

Multiplicative neural language models

- Suppose we have auxiliary vector $\mathbf{u}$ e.g. an image embedding
- We will model $P(w_n|w_{1:n-1}, \mathbf{u})$ by finding F latent factors to explain the multimodal embedding space
- Let $\mathbf{T} \in \mathcal{R}^{V \times K \times G}$ be a tensor, where V is vocabulary size, K is word embedding dimension, G is the dimension of $\mathbf{u}$ i.e. the number of slices of $\mathbf{T}$
- We can model $\mathbf{T}$ as a tensor factorizable into three matrices (where $\mathbf{W}^{ij} \in \mathcal{R}^{I \times J}$)

$$T_u = (\mathbf{W}^{fv})^T \cdot diag(\mathbf{W}^{fg}\mathbf{u}) \cdot \mathbf{W}^{fk}$$

- By multiplying the two outer matrices from above, we get $\mathbf{E} = (\mathbf{W}^{fk})^T \cdot \mathbf{W}^{fv}$, a word embedding matrix independent of u

Multiplicative neural language models

- As in the LBL, we predict the next word representation with

$$\hat{r} = \sum_{i=1}^{n-1} C_i \mathbf{E}_{w_i}$$

where $\mathbf{E}_{w_i}$ is word w_i's embedding, and C_i is a context matrix

- We use a softmax to get a probability distribution

$$P(w_n = i | w_{1:n-1}, \mathbf{u}) = \frac{\exp(\mathbf{W}^{fv}(:, i) f + b_i)}{\sum_j^V \exp(\mathbf{W}^{fv}(:, j) f + b_j)}$$

where factor outputs $f = (\mathbf{W}^{fk} \hat{r}) \cdot (\mathbf{W}^{fg} u)$ depend on u

- Effectively, this model replaces the word embedding matrix R from the LBL with the tensor $\mathbf{T}$, which depends on $\mathbf{u}$

Structure-Content Neural Language Models

- This model, proposed by Kiros et al. is a form of multiplicative neural language model
- We condition on a vector $\mathbf{v}$, as above
- However, $\mathbf{v}$ is an additive function of "content" and "structure" vectors
  - The content vector $\mathbf{u}$ may be an image embedding
  - The structure vector $\mathbf{t}$ is an input series of POS tags
- We are modelling $P(w_n | w_{1:n-1}, \mathbf{t}_{n:n+k}, \mathbf{u})$
  - Previous words and future structure

A bicycle ________ (IN DT NN - -)

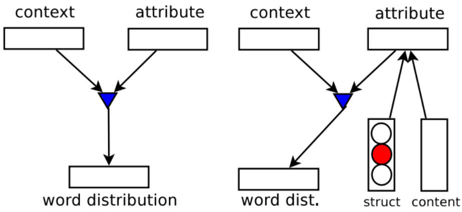
VBN

Structure-Content Neural Language Models

- We can predict a vector $\hat{v}$ of combined structure and content information (the T's are context matrices)

$$\hat{v} = \max\left(\sum_{n}^{n+k}(T^{(i)}t_i) + T_u\mathbf{u} + b, 0\right)$$

- We continue as with the multiplicative model described above
- Note that the content vector u can represent an image or a sentence - using a sentence embedding as u, we can learn on text alone



(a) Multiplicative NLM (b) Structure-content NLM

Caption Generation

1. Embed image
2. Use image embedding and closest images/sentences in dataset to make bag of concepts
3. Get set of all "medium-length" POS sequences
4. Sample a concept conditioning vector and a POS sequence
5. Compute MAP estimate from SC-NLM
6. Generate 1000 descriptions, rank top 5 using scoring function
   - Embed description
   - Get cosine similarity between sentence and image embeddings
   - Kneser-Ney trigram model trained on large corpus - compute log-prob of sentence
   - Average the cosine similarity and the trigram model scores

Experiments: Retrieval

- Trained on Flickr8K/Flickr30K
- Each image has 5 caption sentences
- Metric is Recall-K - how often is correct caption returned in top K results? (or vice versa)
- Best results are state-of-the-art, using OxfordNet features

Flickr8K								
	Image Annotation				Image Search			
Model	**R@1**	**R@5**	**R@10**	**Med** r	**R@1**	**R@5**	**R@10**	**Med** r
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
SDT-RNN [6]	4.5	18.0	28.6	32	6.1	18.5	29.0	29
† DeViSE [5]	4.8	16.5	27.3	28	5.9	20.1	29.6	29
† SDT-RNN [6]	6.0	22.7	34.0	23	6.6	21.6	31.7	25
DeFrag [15]	5.9	19.2	27.3	34	5.2	17.6	26.5	32
† DeFrag [15]	12.6	32.9	44.0	14	9.7	29.6	42.5	15
m-RNN [7]	<u>14.5</u>	<u>37.2</u>	<u>48.5</u>	<u>11</u>	11.5	<u>31.0</u>	42.4	15
Our model	13.5	36.2	45.7	13	10.4	<u>31.0</u>	<u>43.7</u>	<u>14</u>
Our model (OxfordNet)	**18.0**	**40.9**	**55.0**	**8**	**12.5**	**37.0**	**51.5**	**10**

Figure 7: Flickr8K retrieval results

Experiments: Retrieval

- Trained on Flickr8K/Flickr30K
- Each image has 5 caption sentences
- Metric is Recall-K - how often is correct caption returned in top K results? (or vice versa)
- Best results are state-of-the-art, using OxfordNet features

Flickr30K								
	Image Annotation				Image Search			
Model	**R@1**	**R@5**	**R@10**	**Med** ***r***	**R@1**	**R@5**	**R@10**	**Med** ***r***
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
† DeViSE [5]	4.5	18.1	29.2	26	6.7	21.9	32.7	25
† SDT-RNN [6]	9.6	29.8	41.1	16	8.9	29.8	41.1	16
† DeFrag [15]	14.2	37.7	51.3	10	10.2	30.8	44.2	14
† DeFrag + Finetune CNN [15]	16.4	40.2	54.7	8	10.3	31.4	44.5	13
m-RNN [7]	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Our model	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Our model (OxfordNet)	**23.0**	**50.7**	**62.9**	**5**	**16.8**	**42.0**	**56.5**	**8**

Figure 8: Flickr30K retrieval results

Qualitative Results - Caption Generation Successes

- Generation is difficult to evaluate quantitatively

a car is parked in
the middle of nowhere .

a wooden table and chairs
arranged in a room .

there is a cat sitting on a shelf .

a ferry boat on a marina
with a group of people .

a little boy with a bunch
of friends on the street .

Qualitative Results - Caption Generation Failures

- Generation is difficult to evaluate quantitatively

the two birds are trying to be seen in the water . (can't count)

a giraffe is standing next to a fence in a field . (hallucination)

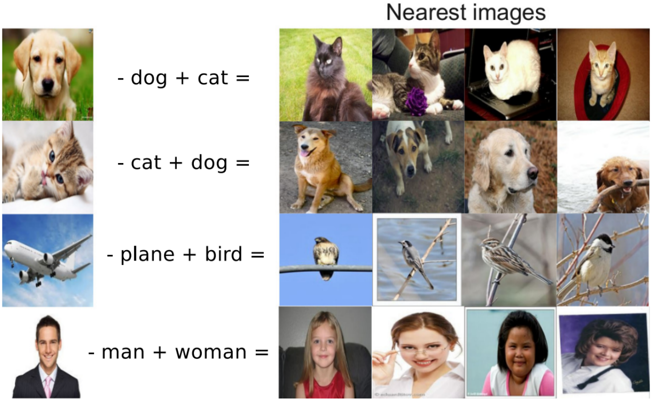
a parked car while driving down the road . (contradiction)

the handlebars are trying to ride a bike rack . (nonsensical)

a woman and a bottle of wine in a garden . (gender)

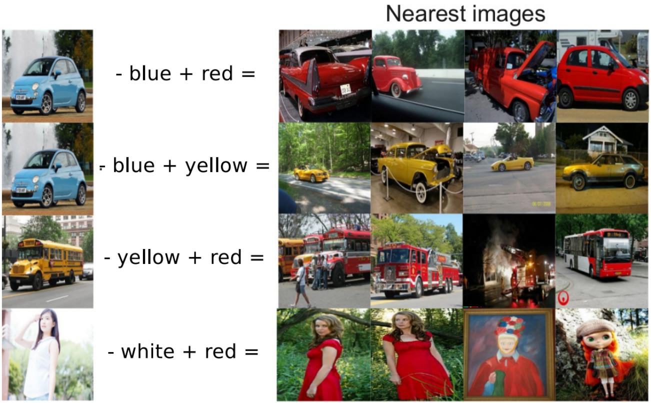
Qualitative Results - Analogies

- We can do analogical reasoning, modelling an image as roughly the sum of its components



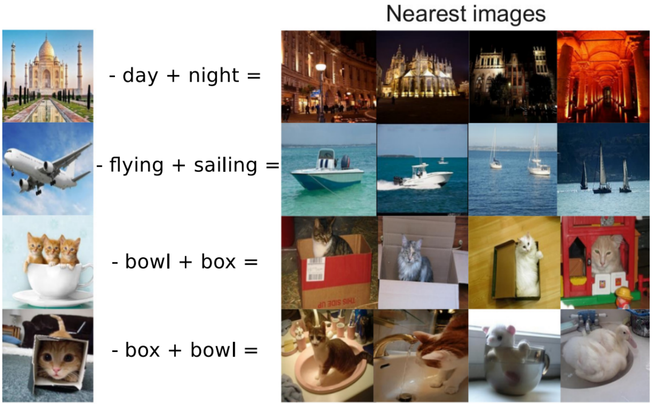
Qualitative Results - Analogies

- We can do analogical reasoning, modelling an image as roughly the sum of its components



Qualitative Results - Analogies

- We can do analogical reasoning, modelling an image as roughly the sum of its components



Conclusions

- In their paper, Kiros et al. present a model for image captioning and retrieval
- The model is inspired by translation systems, and aims to jointly embed images and their captions in the same space
- To decode from the representation space, we condition on an auxiliary content vector (such as an image or sentence representation) and a structure vector (such as POS tags)
- Since the publication of this paper, advances have been made on related problems, such as:
  - Image generation from a given caption
  - Attention-based captioning
  - State of the art caption generation on the MS-COCO dataset are Google's model (Show and Tell: A Neural Image Caption Generator, 2015) and MSR's model (From Captions to Visual Concepts and Back, 2015) with 32% of captions passing the Turing test, compared to 16% for this model

Questions?

Thanks for your attention!