CS 2770: Computer Vision

Vision, Language, Reasoning

Prof. Adriana Kovashka
University of Pittsburgh
March 5, 2019

Plan for this lecture

- Image captioning
  - Tool: Recurrent neural networks
  - Captioning for video
  - Diversifying captions
- Visual-semantic spaces
- Visual question answering
  - Incorporating knowledge and reasoning
  - Tool: Graph convolutional networks

Motivation: Descriptive Text for Images

"It was an arresting face, pointed of chin, square of jaw. Her eyes were pale green without a touch of hazel, starred with bristly black lashes and slightly tilted at the ends. Above them, her thick black brows slanted upward, cutting a startling oblique line in her magnolia-white skin–that skin so prized by Southern women and so carefully guarded with bonnets, veils and mittens against hot Georgia suns"

Scarlett O'Hara described in Gone with the Wind

Some pre-RNN good results

This is a picture of one sky, one road and one sheep. The gray sky is over the gray road. The gray sheep is by the gray road.

Here we see one road, one sky and one bicycle. The road is near the blue sky, and near the colorful bicycle. The colorful bicycle is within the blue sky.

This is a picture of two dogs. The first dog is near the second furry dog.

Kulkarni et al., CVPR 2011

Some pre-RNN bad results

Missed detections:

Here we see one potted plant.

This is a picture of one dog.

False detections:

There are one road and one cat. The furry road is in the furry cat.

This is a picture of one tree, one road and one person. The rusty tree is under the red road. The colorful person is near the rusty tree, and under the red road.

Incorrect attributes:

This is a photograph of two sheeps and one grass. The first black sheep is by the green grass, and by the second black sheep. The second black sheep is by the green grass.

This is a photograph of two horses and one grass. The first feathered horse is within the green grass, and by the second feathered horse. The second feathered horse is within the green grass.

Kulkarni et al., CVPR 2011

Results with Recurrent Neural Networks

"man in black shirt is playing guitar."

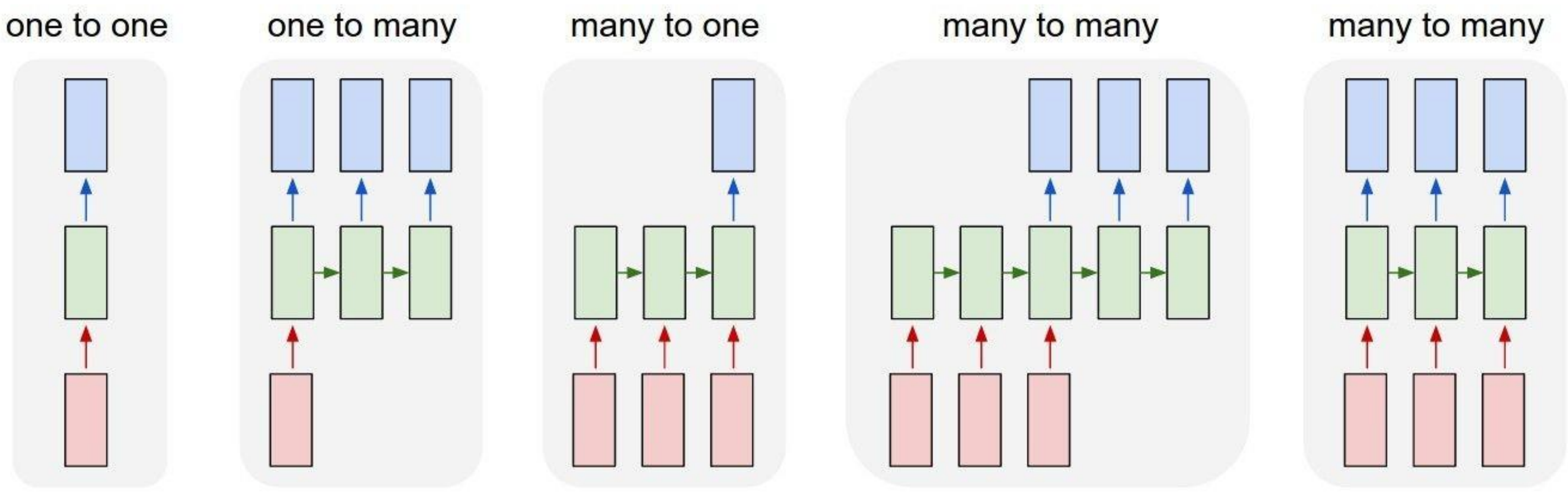
"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

Karpathy and Fei-Fei, CVPR 2015

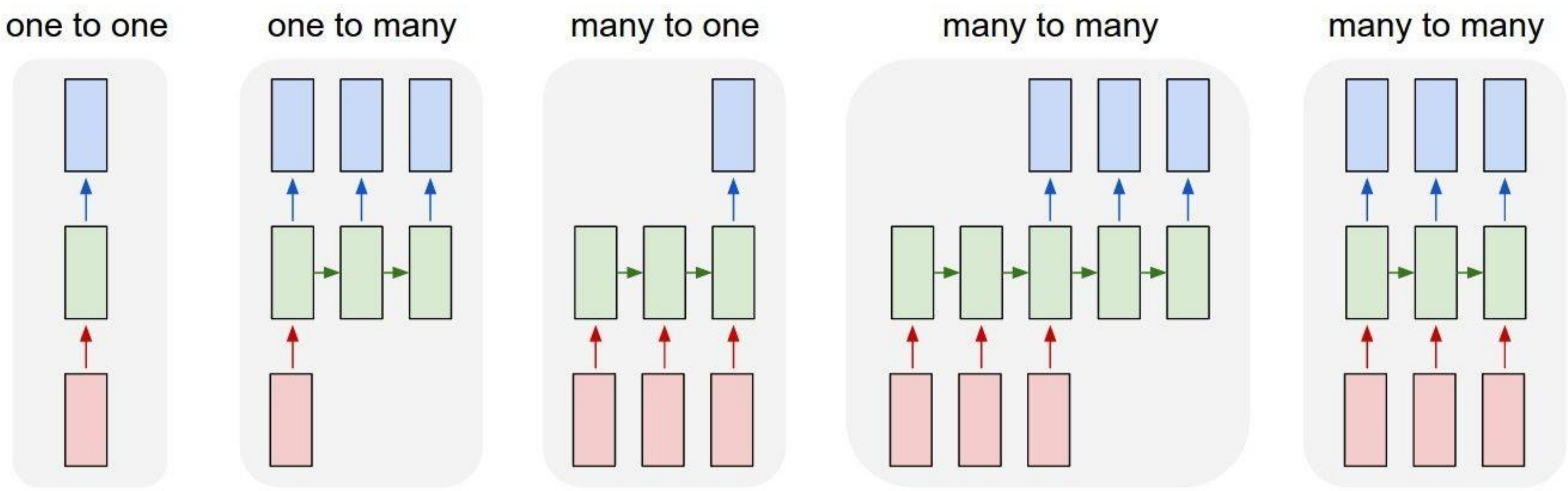
Recurrent Networks offer a lot of flexibility:



vanilla neural networks

Andrej Karpathy

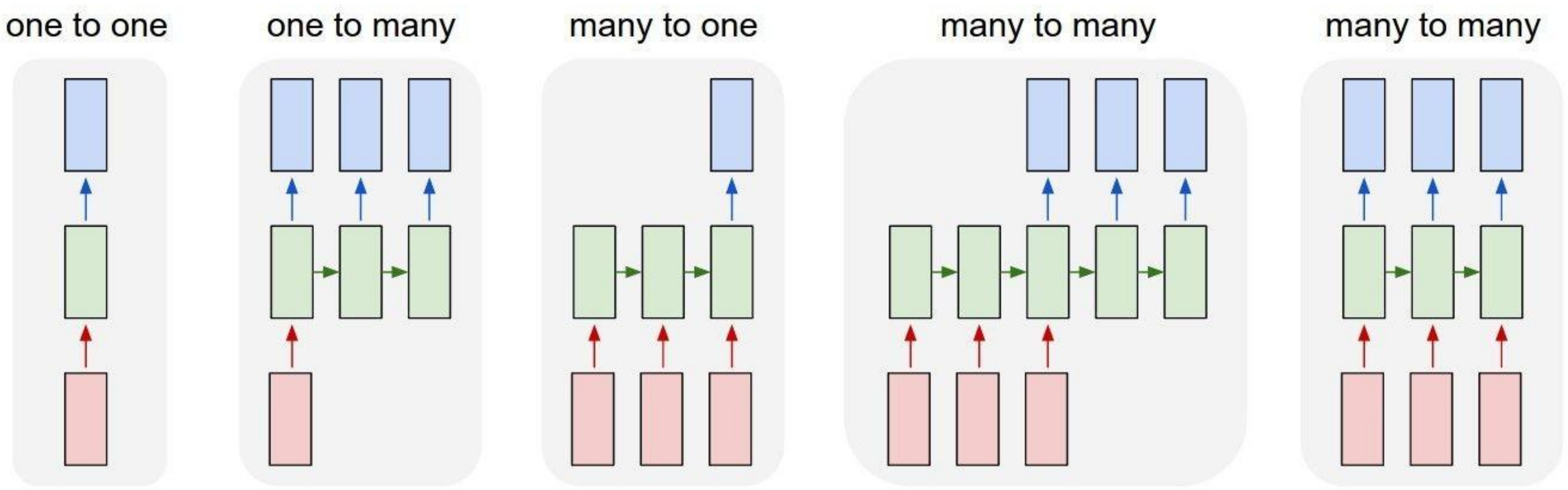
Recurrent Networks offer a lot of flexibility:



e.g. **image captioning**
image -> sequence of words

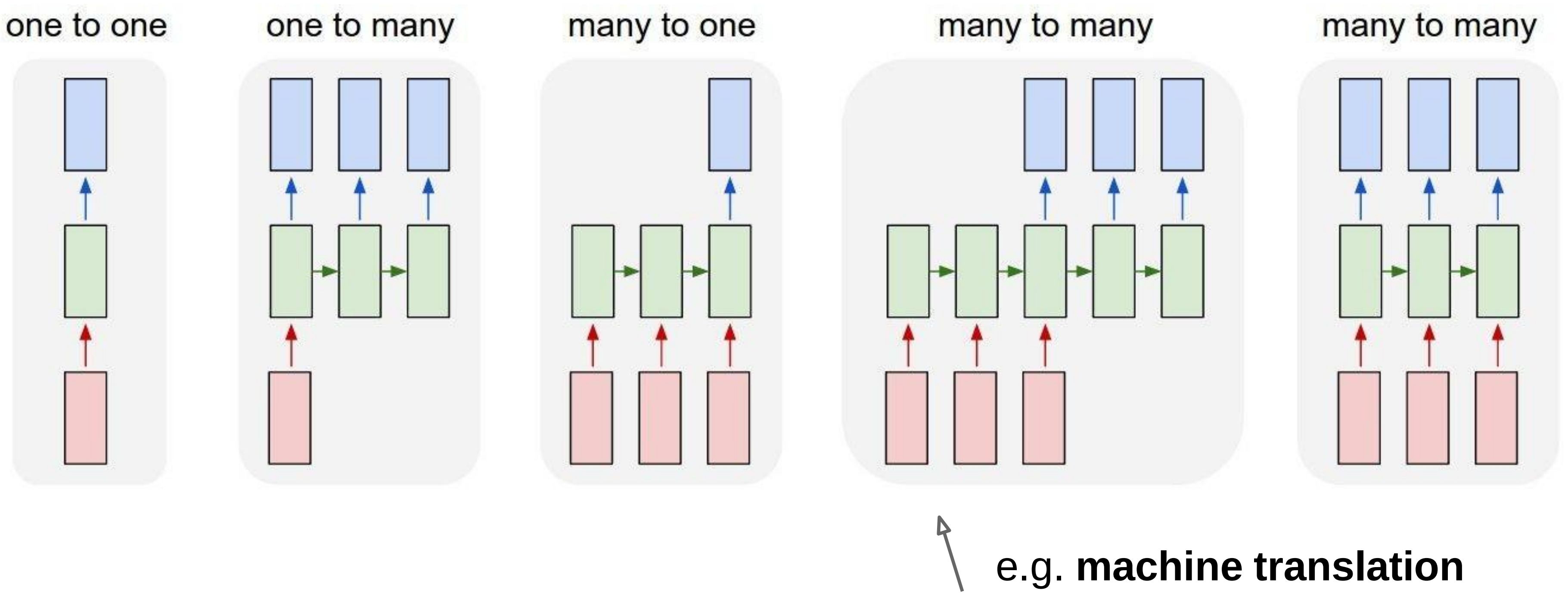
Andrej Karpathy

Recurrent Networks offer a lot of flexibility:



e.g. **sentiment classification**
sequence of words -> sentiment

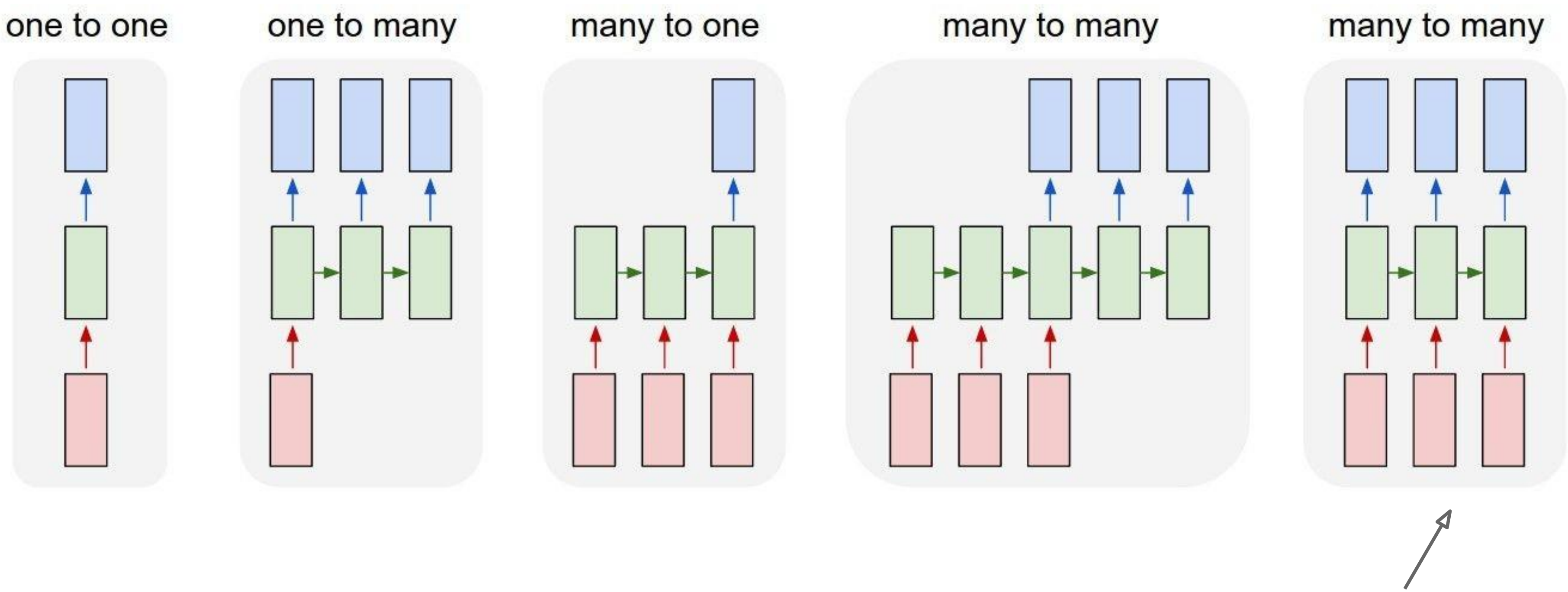
Recurrent Networks offer a lot of flexibility:



e.g. **machine translation**
seq of words -> seq of words

Andrej Karpathy

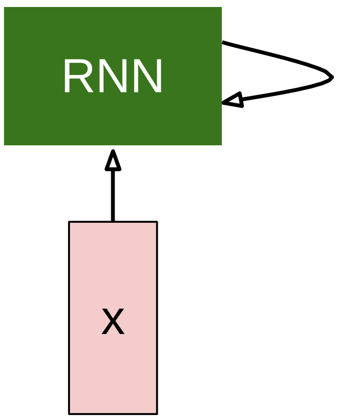
Recurrent Networks offer a lot of flexibility:



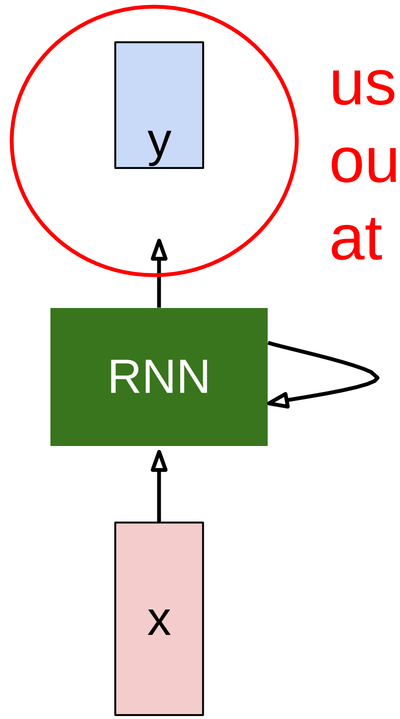
e.g. **video classification on frame level**

Andrej Karpathy

Recurrent Neural Network



Recurrent Neural Network



usually want to
output a prediction
at some time steps

Adapted from Andrej Karpathy

Recurrent Neural Network

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

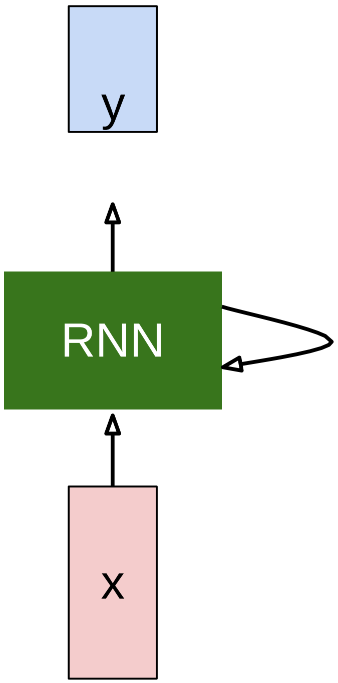
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

old state

input vector at some time step

some function with parameters W

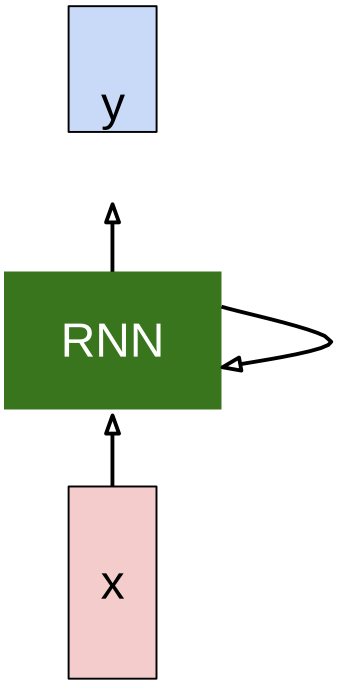


Recurrent Neural Network

We can process a sequence of vectors **x** by applying a recurrence formula at every time step:

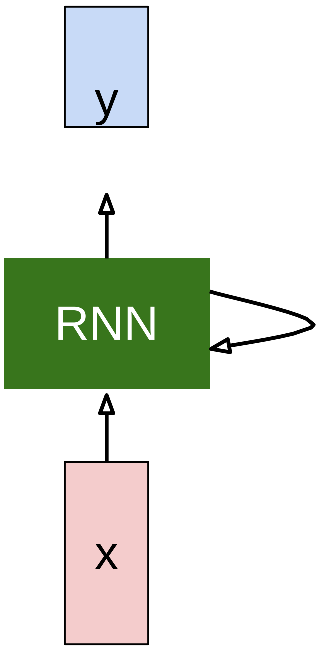
$$h_t = f_W(h_{t-1}, x_t)$$

Notice: the same function and the same set of parameters are used at every time step.



(Vanilla) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

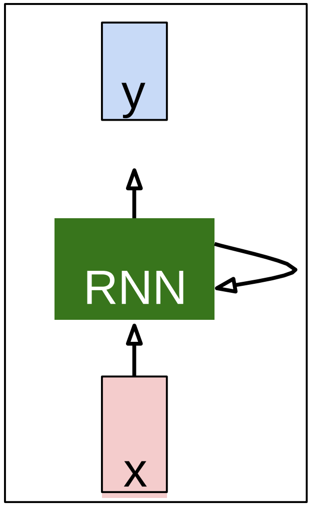
$$y_t = W_{hy}h_t$$

Example

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

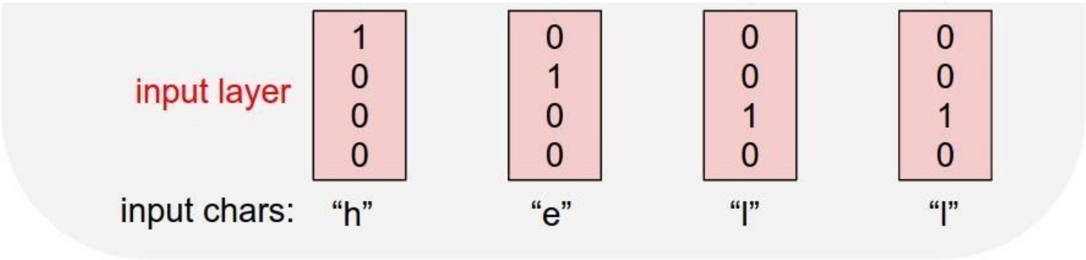


Example

Character-level language model example

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”



Andrej Karpathy

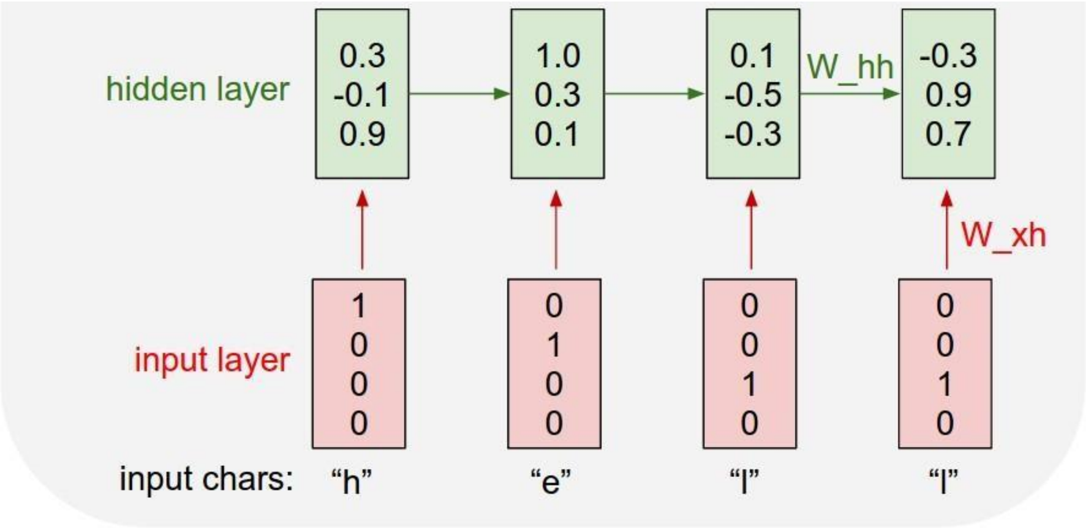
Example

Character-level language model example

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

Vocabulary: [h,e,l,o]

Example training sequence: **“hello”**

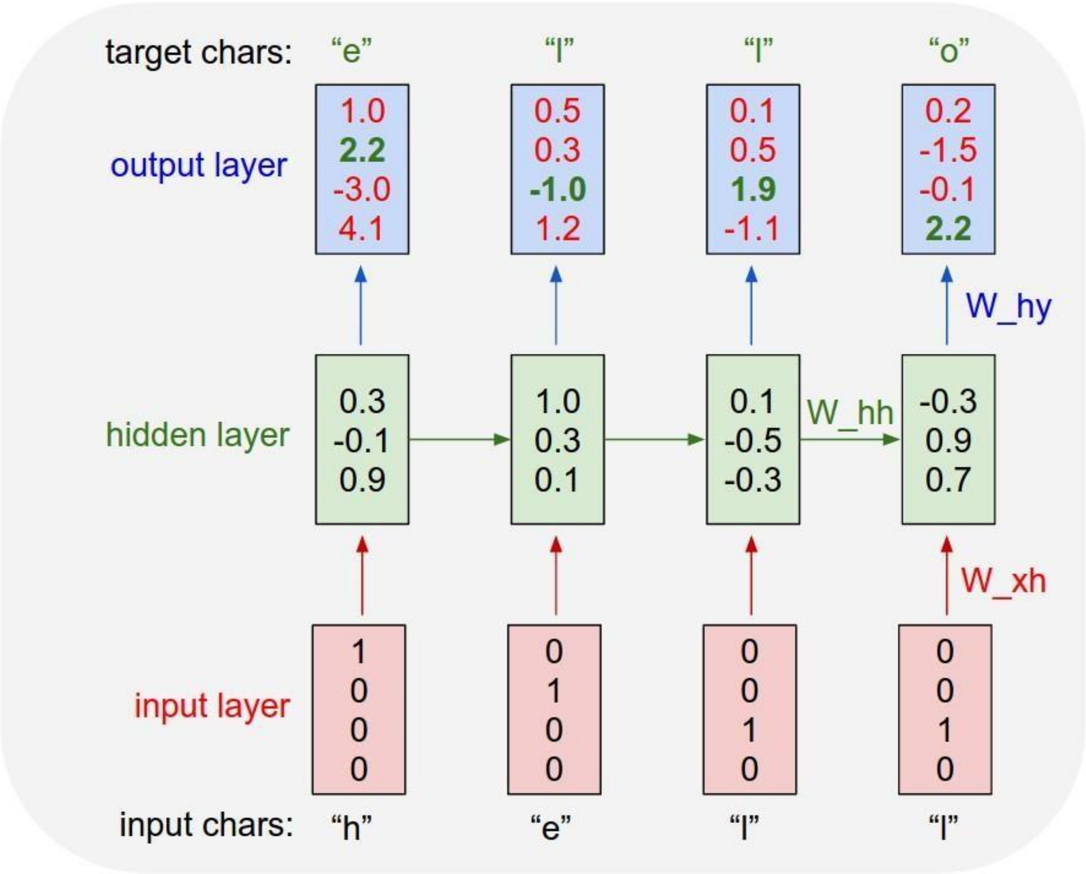


Example

Character-level language model example

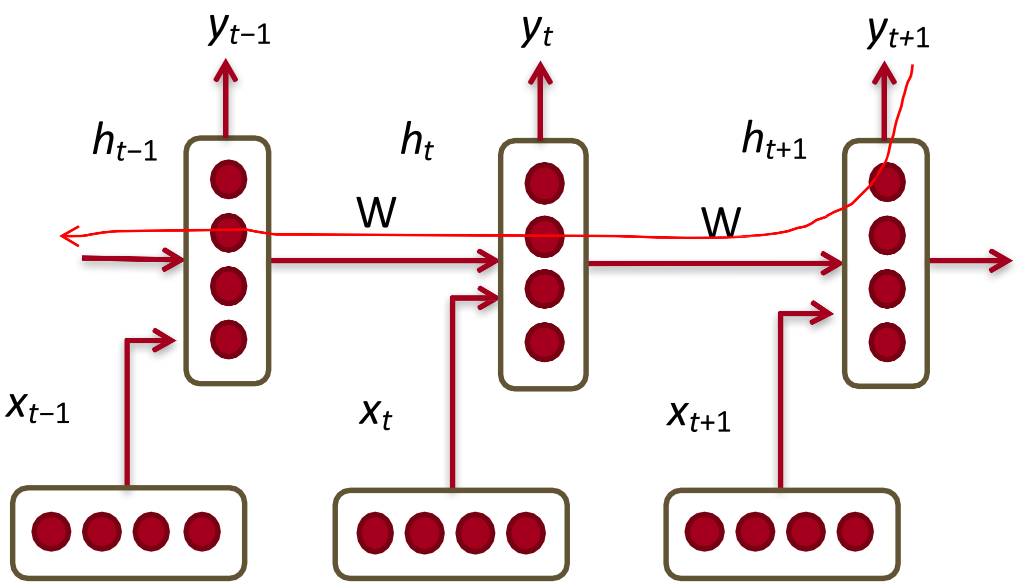
Vocabulary:
[h,e,l,o]

Example training sequence:
"hello"



The vanishing gradient problem

- The error at a time step ideally can tell a previous time step from many steps away to change during backprop
- But we're multiplying together many values between 0 and 1



Adapted from Richard Socher

The vanishing gradient problem

- Total error is the sum of each error at time steps t

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial W}$$

- Chain rule:

$$\frac{\partial E_t}{\partial W} = \sum_{k=1}^{t} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

- More chain rule:

$$\frac{\partial h_t}{\partial h_k} = \prod_{j=k+1}^{t} \frac{\partial h_j}{\partial h_{j-1}}$$

- Derivative of vector wrt vector is a Jacobian matrix of partial derivatives; norm of this matrix can become very small or very large quickly [Bengio et al 1994], leading to vanishing/exploding gradient

Adapted from Richard Socher

The vanishing gradient problem for language models

- In the case of language modeling or question answering words from time steps far away are not taken into consideration when training to predict the next word

- Example:

  Jane walked into the room. John walked in too. It was late in the day. Jane said hi to _____

Gated Recurrent Units (GRUs)

- More complex hidden unit computation in recurrence!
- Introduced by Cho et al. 2014
- Main ideas:
  - keep around memories to capture long distance dependencies
  - allow error messages to flow at different strengths depending on the inputs

Gated Recurrent Units (GRUs)

- Standard RNN computes hidden layer at next time step directly: $$h_t = f\left(W^{(hh)}h_{t-1} + W^{(hx)}x_t\right)$$
- GRU first computes an update **gate** (another layer) based on current input word vector and hidden state

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$

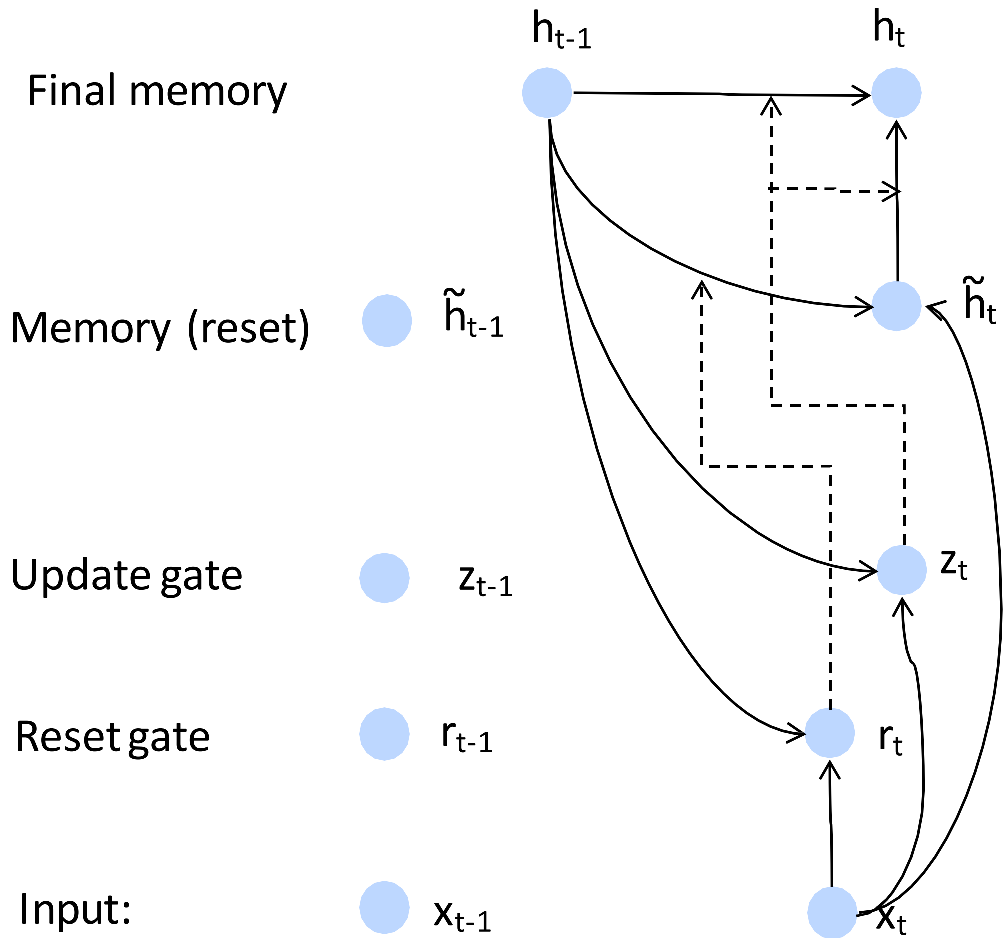
- Compute reset gate similarly but with different weights

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$

Gated Recurrent Units (GRUs)

- Update gate $$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$
- Reset gate $$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$
- New memory content: $$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$
  If reset gate unit is ~0, then this ignores previous memory and only stores the new word information
- Final memory at time step combines current and previous time steps: $$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

Gated Recurrent Units (GRUs)



$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$

$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$

$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

Gated Recurrent Units (GRUs)

- If reset is close to 0, ignore previous hidden state: Allows model to drop information that is irrelevant in the future

$$z_t = \sigma\left(W^{(z)}x_t + U^{(z)}h_{t-1}\right)$$
$$r_t = \sigma\left(W^{(r)}x_t + U^{(r)}h_{t-1}\right)$$
$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1}\right)$$
$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

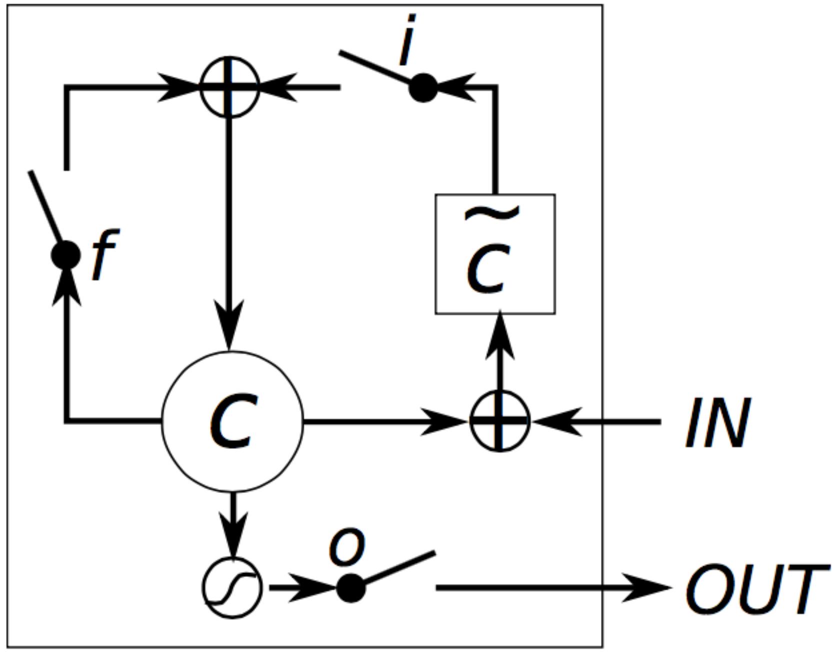
- Update gate z controls how much of past state should matter now
  - If z close to 1, then we can copy information in that unit through many time steps! **Less vanishing gradient**!
- Units with short-term dependencies often have reset gates (r) very active; ones with long-term dependencies have active update gates (z)

Long-short-term-memories (LSTMs)

- Proposed by Hochreiter and Schmidhuber in 1997
- We can make the units even more complex
- Allow each time step to modify
  - Input gate (current cell matters) $i_t = \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1}\right)$
  - Forget (gate 0, forget past) $f_t = \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1}\right)$
  - Output (how much cell is exposed) $o_t = \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1}\right)$
  - New memory cell $\tilde{c}_t = \tanh\left(W^{(c)}x_t + U^{(c)}h_{t-1}\right)$
- Final memory cell: $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$
- Final hidden state: $h_t = o_t \circ \tanh(c_t)$

Adapted from Richard Socher

Long-short-term-memories (LSTMs)



Intuition: memory cells can keep information intact, unless inputs makes them forget it or overwrite it with new input

Cell can decide to output this information or just store it

Generating poetry with RNNs

Sonnet 116 – Let me not …

by William Shakespeare

Let me not to the marriage of true minds
        Admit impediments. Love is not love
Which alters when it alteration finds,
        Or bends with the remover to remove:
O no! it is an ever-fixed mark
        That looks on tempests and is never shaken;
It is the star to every wandering bark,
        Whose worth's unknown, although his height be taken.
Love's not Time's fool, though rosy lips and cheeks
        Within his bending sickle's compass come:
Love alters not with his brief hours and weeks,
        But bears it out even to the edge of doom.
If this be error and upon me proved,
        I never writ, nor no man ever loved.

Andrej Karpathy

Generating poetry with RNNs

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

More info: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Generating poetry with RNNs

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

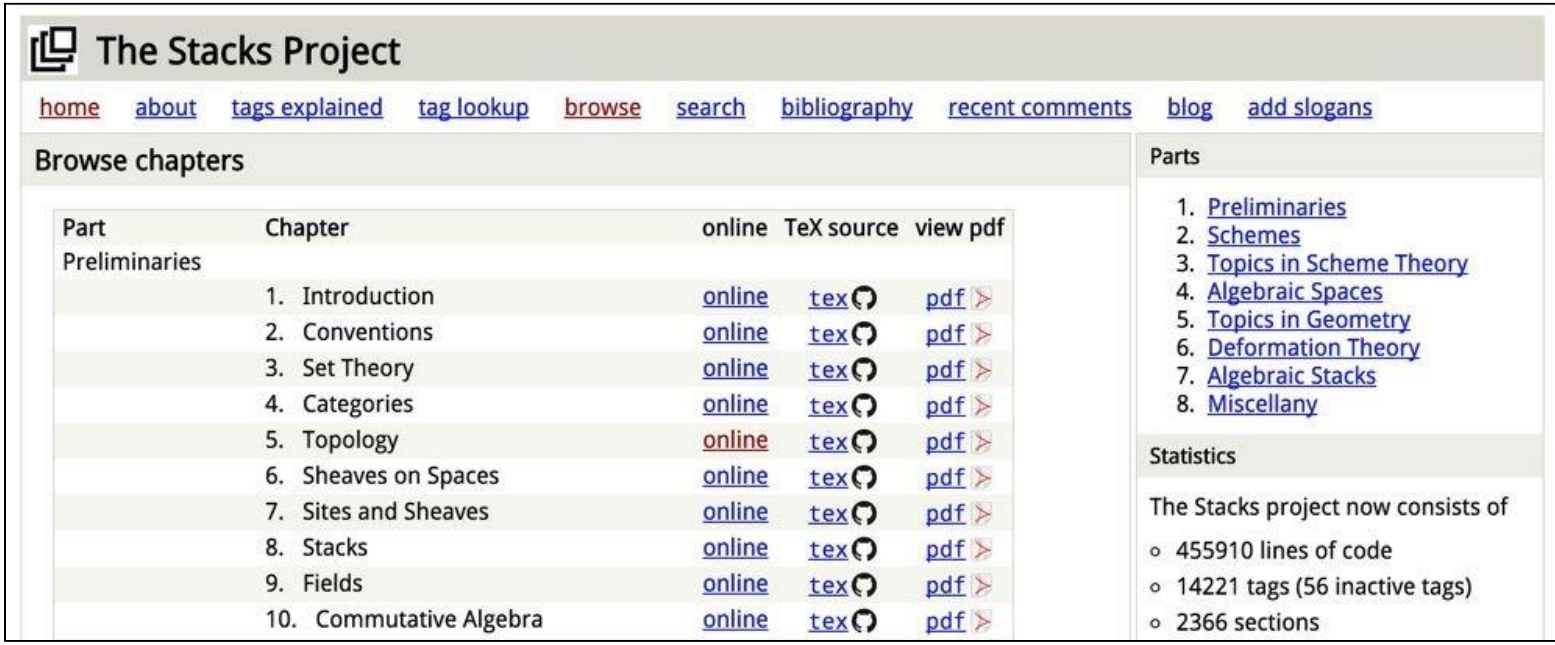
```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

Andrej Karpathy

Generating textbooks with RNNs

open source textbook on algebraic geometry



Latex source

Generating textbooks with RNNs

For $\bigoplus_{n=1,\dots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the comparicoly in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, **??** and the fact that any U affine, see Morphisms, Lemma **??**. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma **??** we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M}^\bullet = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example **??**. It may replace S by $X_{spaces,\acute{e}tale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma **??**. Namely, by Lemma **??** we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ (by the formal open covering X and a single map $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ over U compatible with the complex

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

*When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition **??** (without element is when the closed subschemes are catenary. If T is surjective we may assume that T is connected with residue fields of S. Moreover there exists a closed subspace $Z \subset X$ of X where U in X' is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) *f is locally of finite type. Since $S = \mathrm{Spec}(R)$ and $Y = \mathrm{Spec}(R)$.*

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\dots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\dots,0}$.

Lemma 0.2. *Let X be a locally Noetherian scheme over S, $E = \mathcal{F}_{X/S}$. Set $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. Since $\mathcal{I}^n \subset \mathcal{I}^n$ are nonzero over $i_0 \leq \mathfrak{p}$ is a subset of $\mathcal{J}_{n,0} \circ \overline{A}_2$ works.*

Lemma 0.3. *In Situation **??**. Hence we may assume $\mathfrak{q}' = 0$.*

Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (**??**). On the other hand, by Lemma **??** we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

Andrej Karpathy

Generating textbooks with RNNs

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma **??**. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

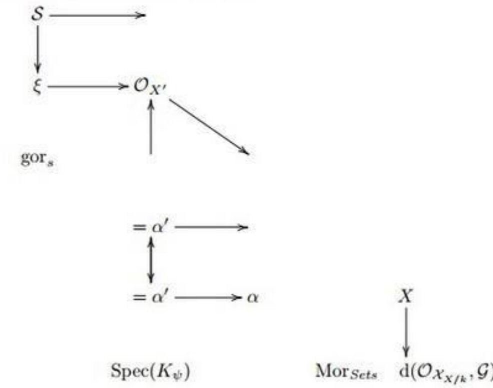
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \ \text{-}1(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1} \mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Andrej Karpathy

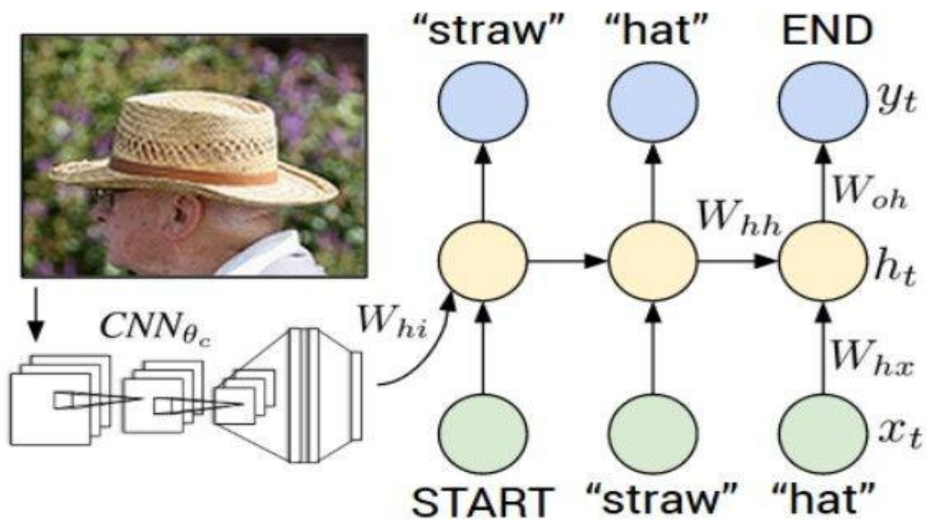
Generating code with RNNs

```c
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

Generated C code

Andrej Karpathy

Image Captioning

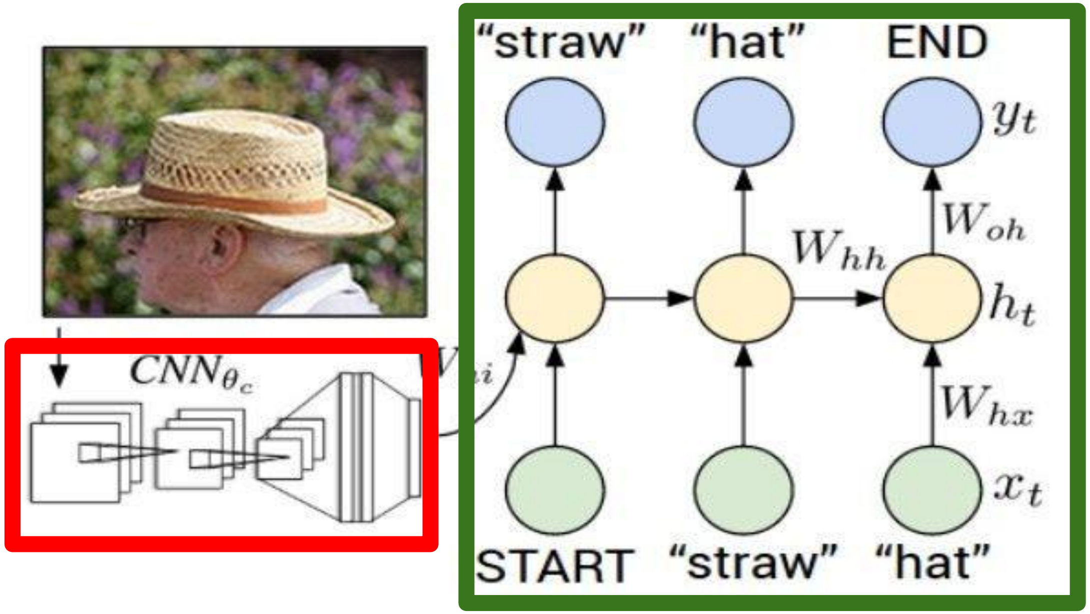


CVPR 2015:
Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
Show and Tell: A Neural Image Caption Generator, Vinyals et al.
Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

Adapted from Andrej Karpathy

Image Captioning

Recurrent Neural Network



Convolutional Neural Network

Image Captioning

test image

test image

Andrej Karpathy

image

conv-64
conv-64
maxpool

conv-128
conv-128
maxpool

conv-256
conv-256
maxpool

conv-512
conv-512
maxpool

conv-512
conv-512
maxpool

FC-4096
FC-4096

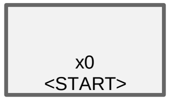
X

Andrej Karpathy

test image

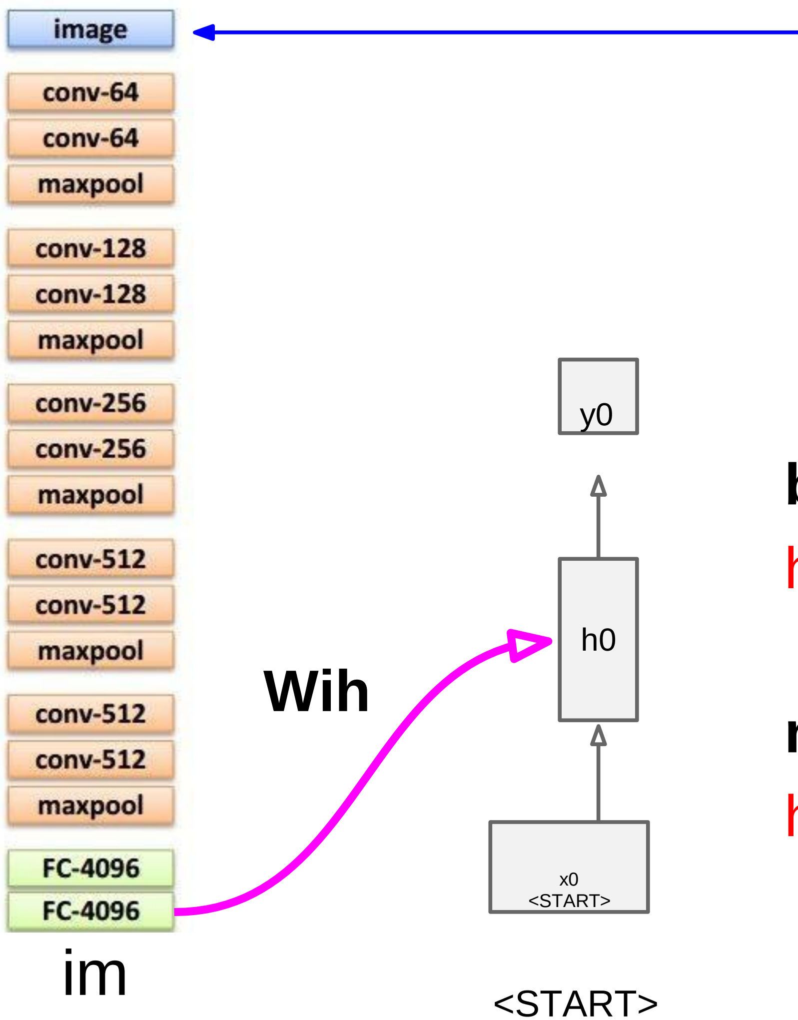
Image Captioning

test image



<START>

Image Captioning



test image

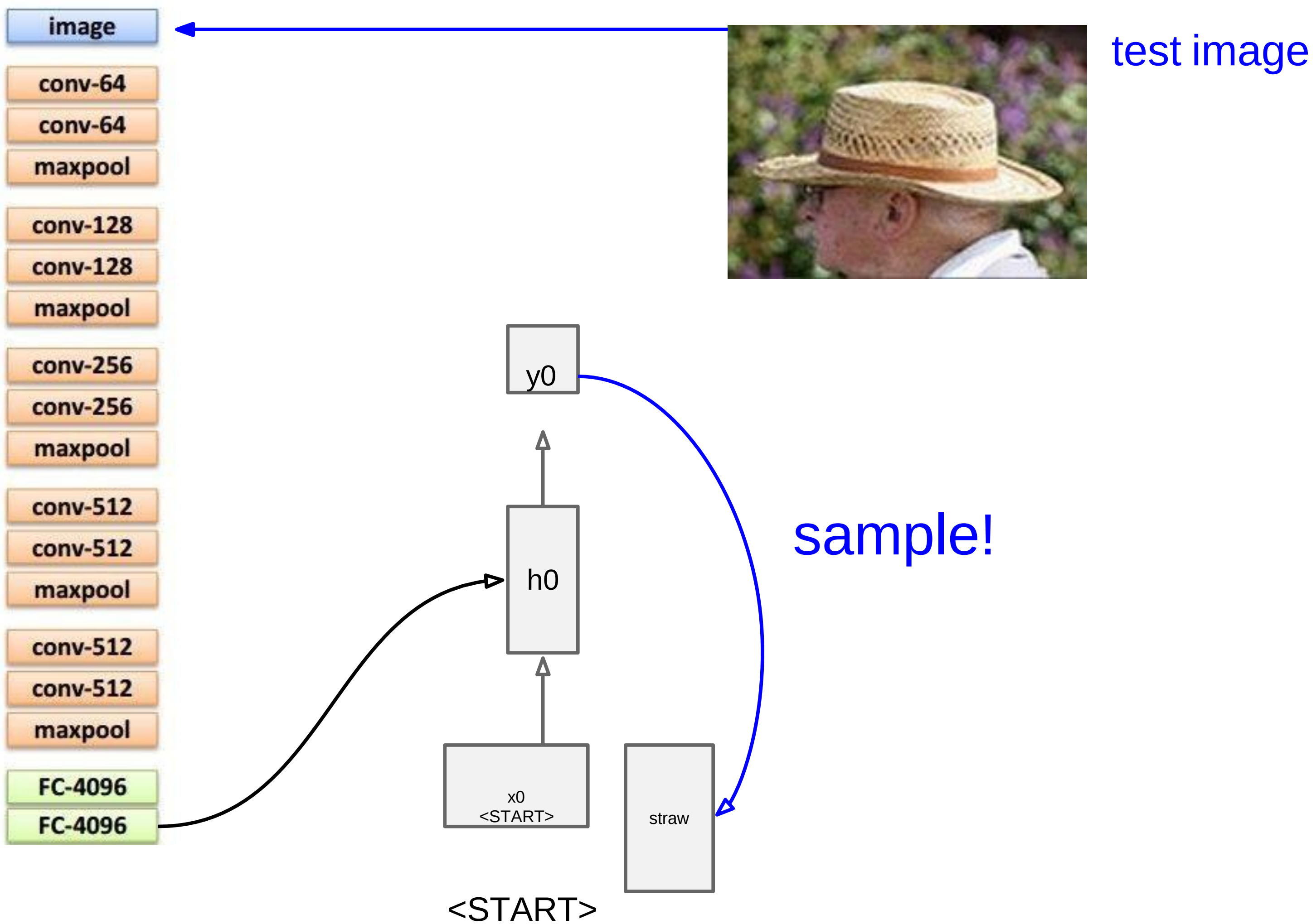
before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

$$h = \tanh(W_{xh} * x + W_{hh} * h + \mathbf{W_{ih} * im})$$

Image Captioning



Andrej Karpathy

Image Captioning

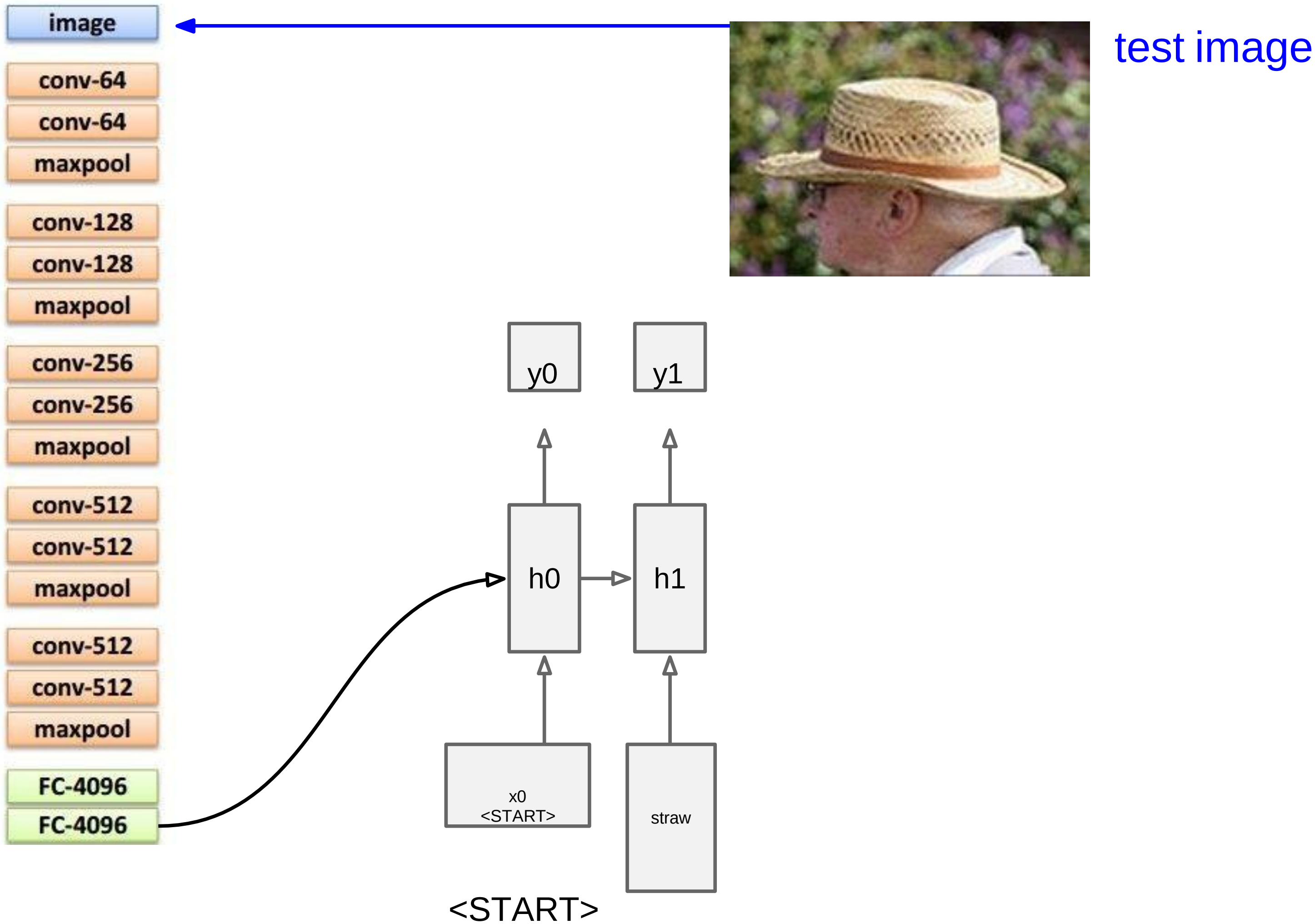
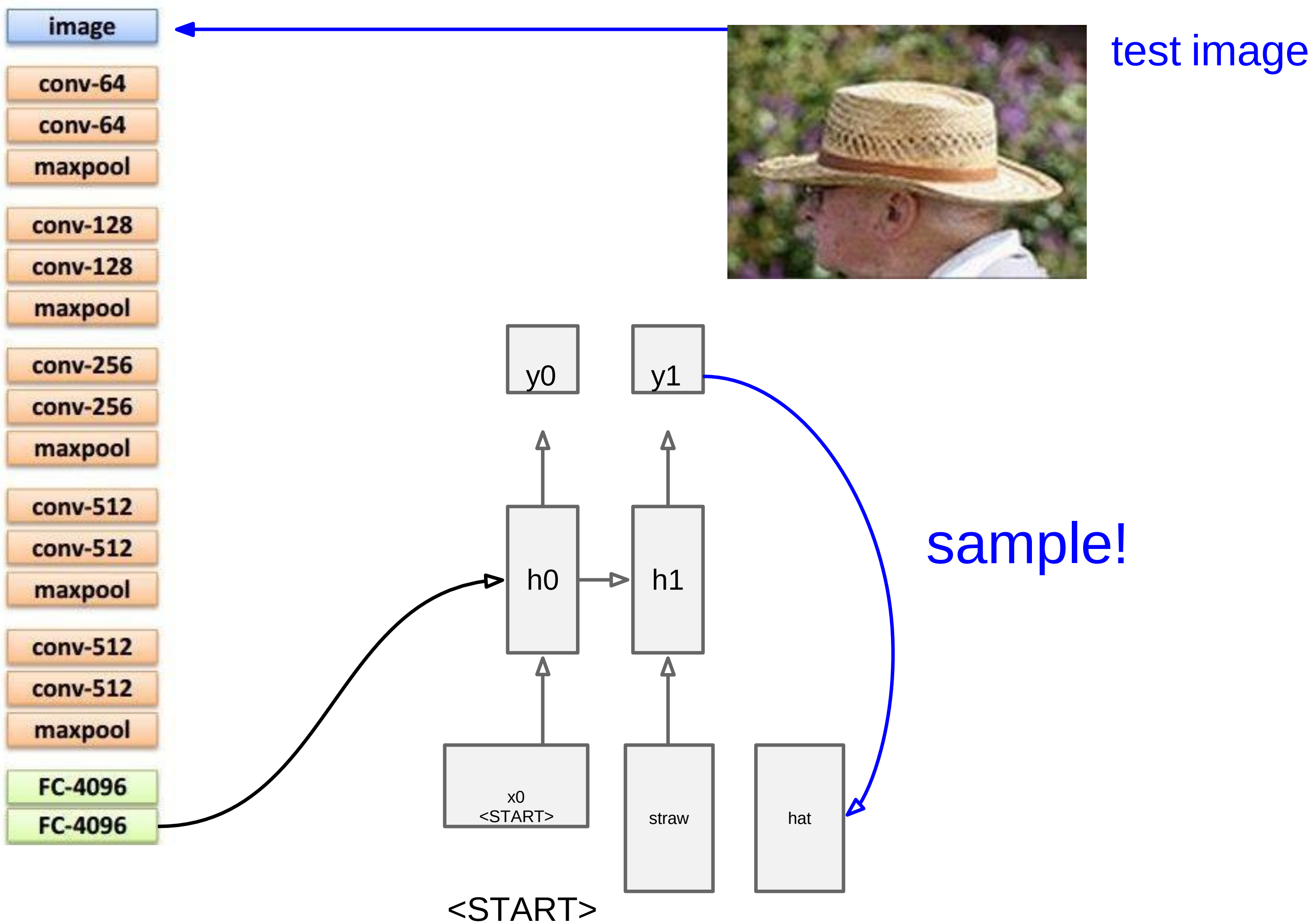


Image Captioning



Andrej Karpathy

Image Captioning

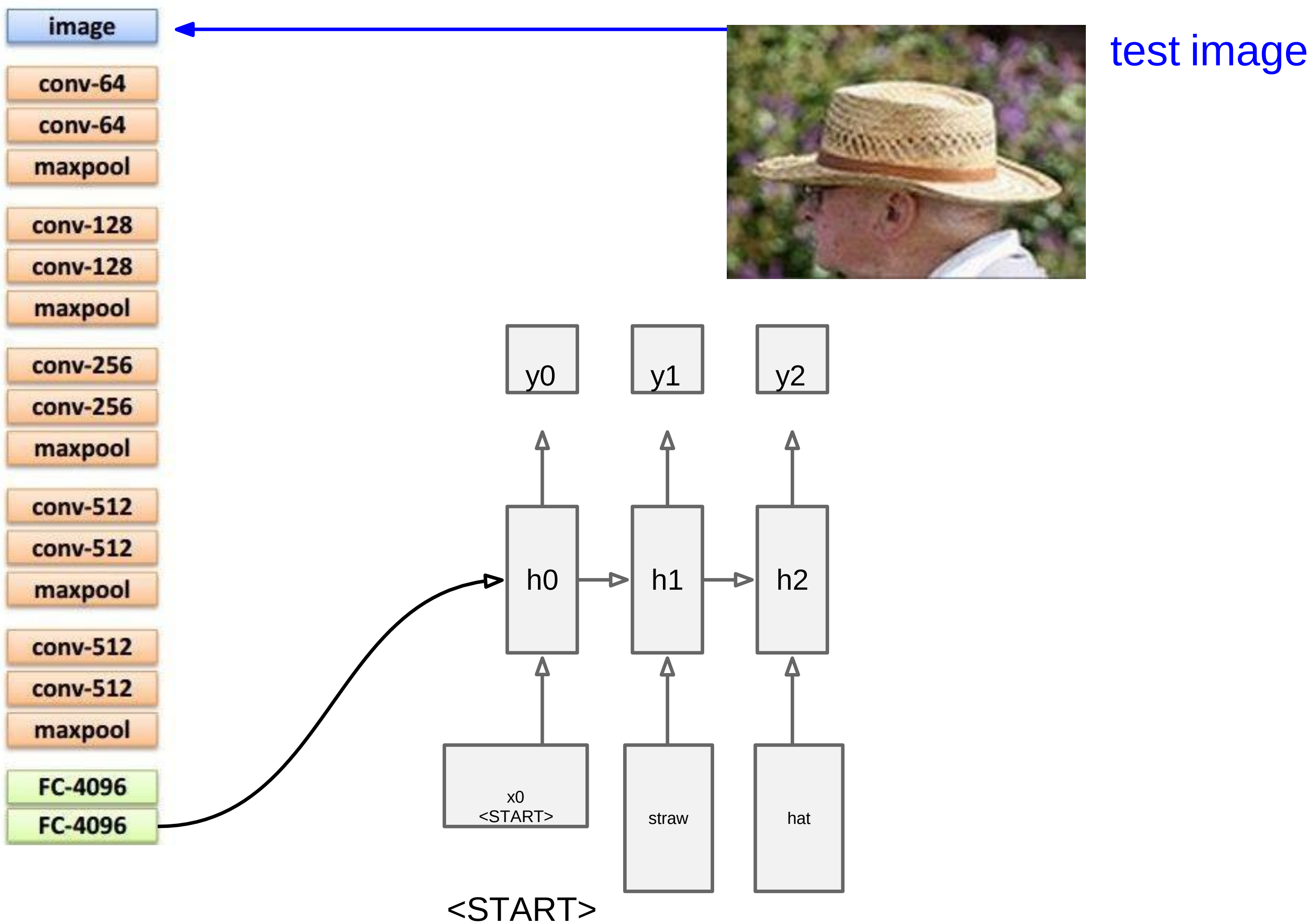
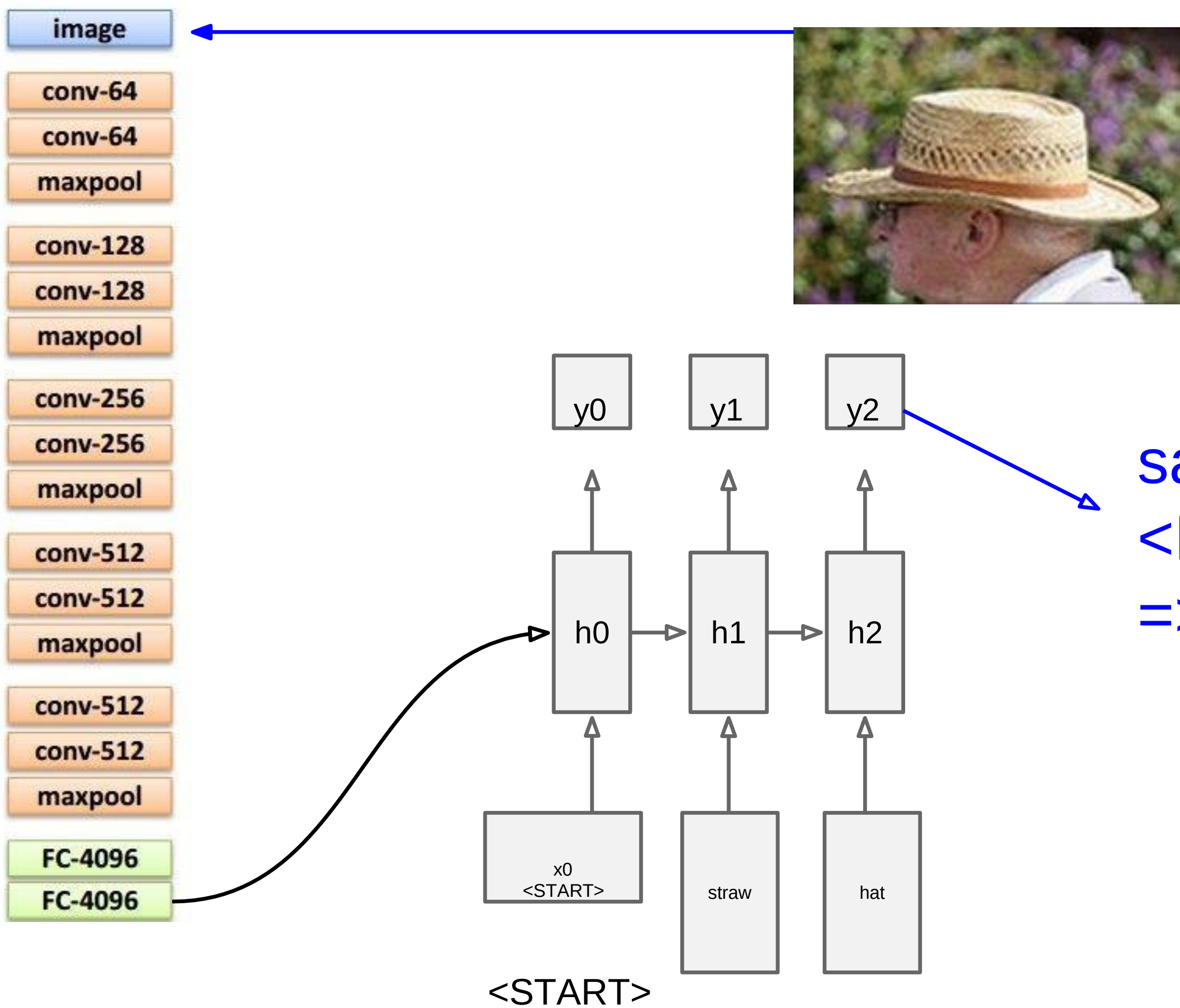


Image Captioning



test image

Caption generated:
“straw hat”

sample
<END> token
=> finish.

Adapted from Andrej Karpathy

Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"a young boy is holding a baseball bat."

"a cat is sitting on a couch with a remote control."

"a woman holding a teddy bear in front of a mirror."

"a horse is standing in the middle of a road."

Plan for this lecture

- Image captioning
  - Tool: Recurrent neural networks
  - Captioning for video
  - Diversifying captions
- Visual-semantic spaces
- Visual question answering
  - Incorporating knowledge and reasoning
  - Tool: Graph convolutional networks

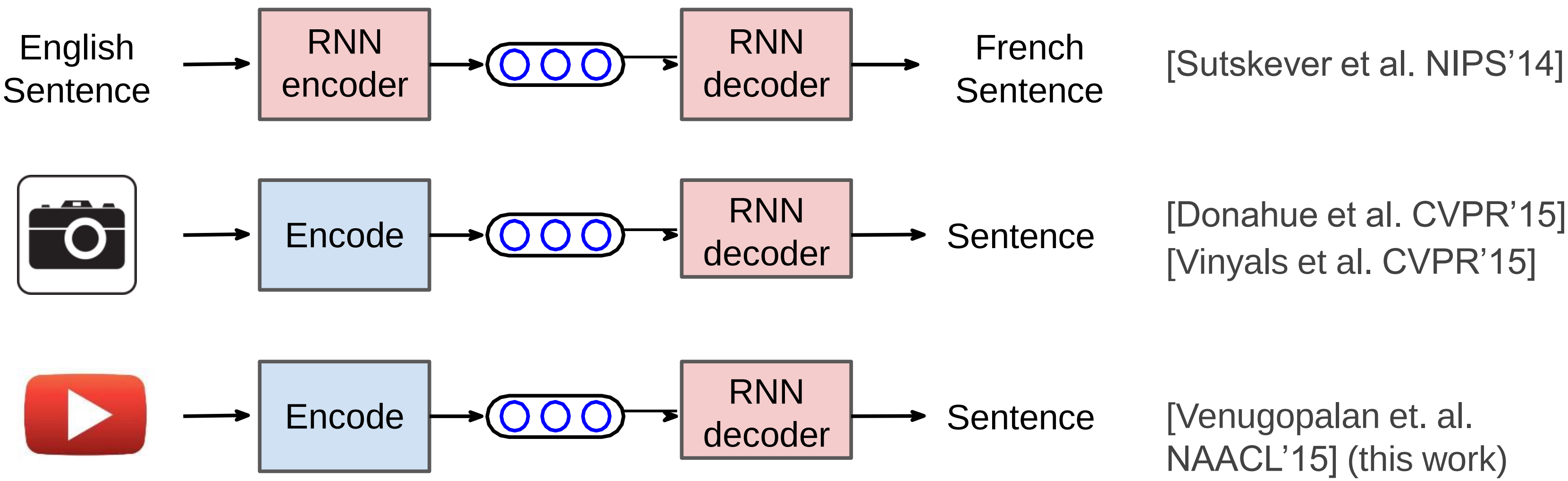
Video Captioning

Generate descriptions for events depicted in video clips

A monkey pulls a dog's tail and is chased by the dog.

Venugopalan et al., "Translating Videos to Natural Language using Deep Recurrent Neural Networks", NAACL-HTL 2015

Video Captioning

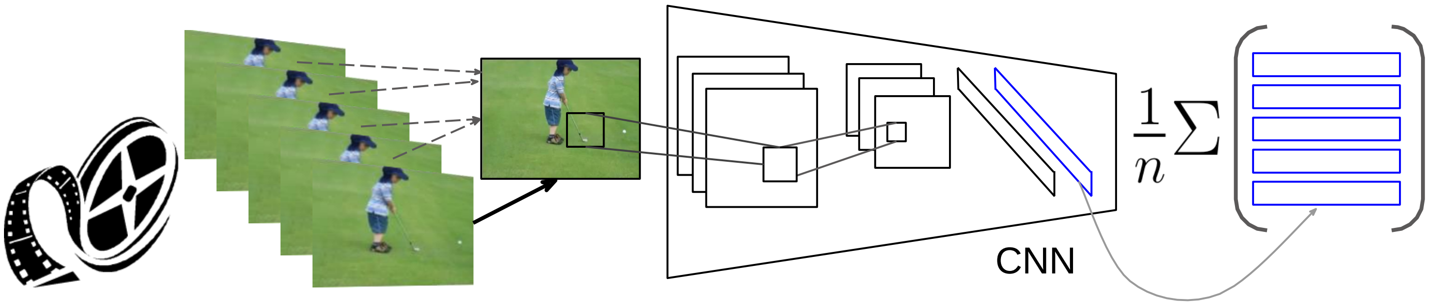


Key Insight:

Generate feature representation of the video and "decode" it to a sentence

Venugopalan et al., "Translating Videos to Natural Language using Deep Recurrent Neural Networks", NAACL-HTL 2015

Video Captioning



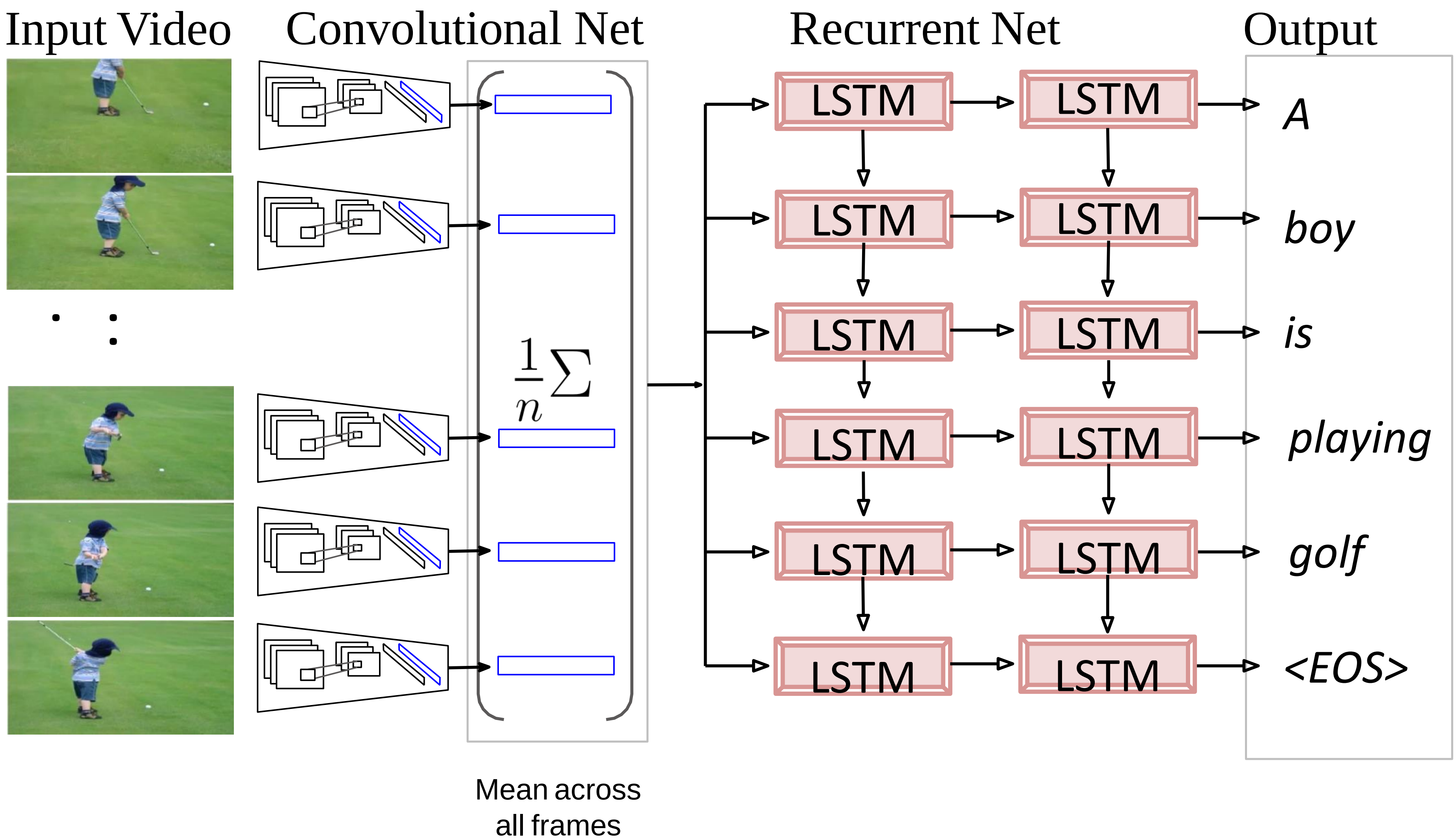
Input Video → Sample frames @1/10

Forward propagate
Output: “fc7” features
(activations before classification layer)

fc7: 4096 dimension
“feature vector”

Venugopalan et al., “Translating Videos to Natural Language using Deep Recurrent Neural Networks”, NAACL-HTL 2015

Video Captioning



Venugopalan et al., “Translating Videos to Natural Language using Deep Recurrent Neural Networks”, NAACL-HTL 2015

Video Captioning

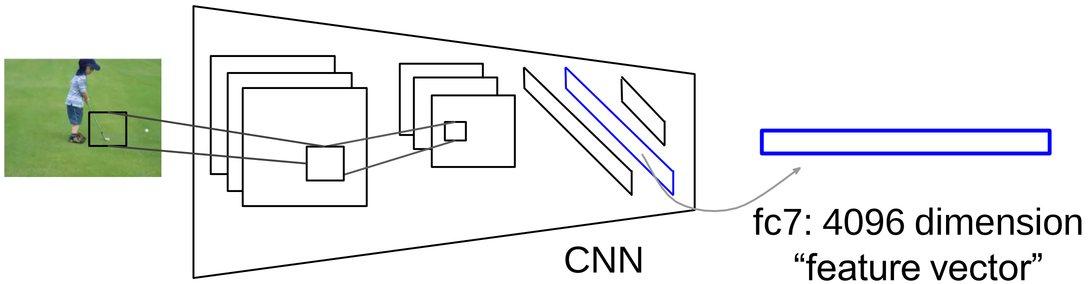
Annotated video data is scarce.

Key Insight:
Use supervised pre-training on data-rich
auxiliary tasks and transfer.

Venugopalan et al., "Translating Videos to Natural Language using Deep Recurrent Neural Networks", NAACL-HTL 2015

Video Captioning

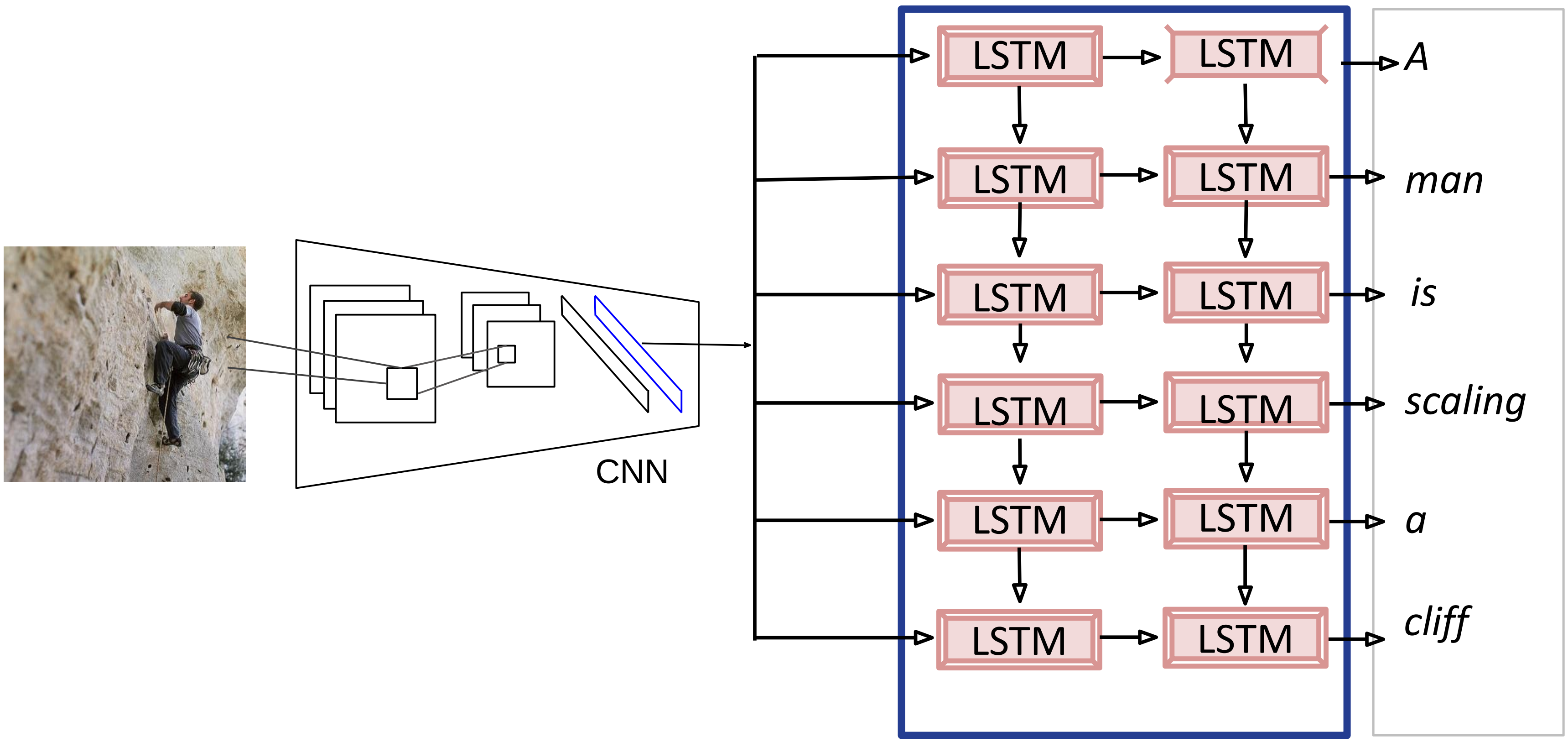
CNN pre-training



- Caffe Reference Net - variation of Alexnet [Krizhevsky et al. NIPS’12]
- 1.2M+ images from ImageNet ILSVRC-12 [Russakovsky et al.]
- Initialize weights of our network.

Video Captioning

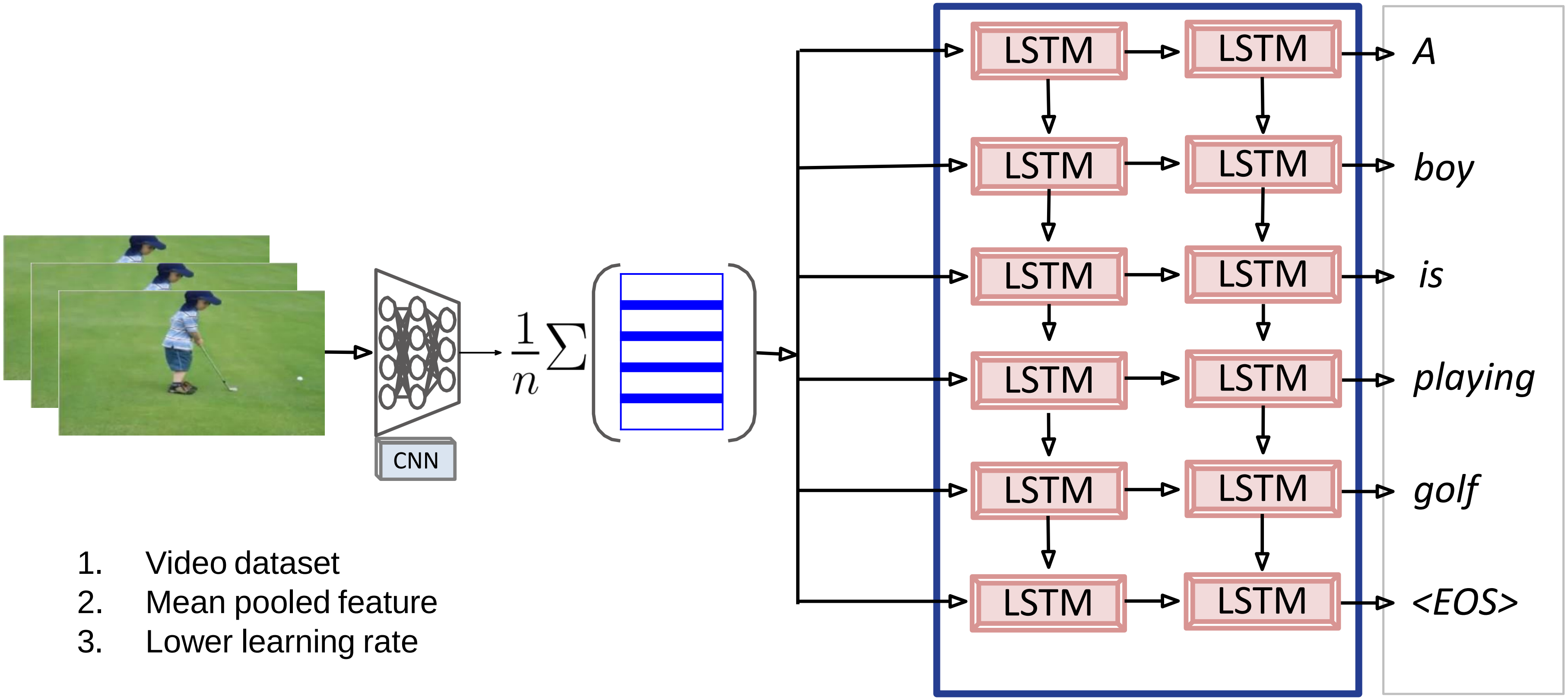
Image-Caption pre-training



Venugopalan et al., “Translating Videos to Natural Language using Deep Recurrent Neural Networks”, NAACL-HTL 2015

Video Captioning

Fine-tuning



1. Video dataset
2. Mean pooled feature
3. Lower learning rate

Venugopalan et al., "Translating Videos to Natural Language using Deep Recurrent Neural Networks", NAACL-HTL 2015

Video Captioning

- A man appears to be plowing a rice field with a plow being pulled by two oxen.
- A man is plowing a mud field.
- Domesticated livestock are helping a man plow.
- A man leads a team of oxen down a muddy path.
- A man is plowing with some oxen.
- A man is tilling his land with an ox pulled plow.
- Bulls are pulling an object.
- Two oxen are plowing a field.
- The farmer is tilling the soil.
- A man in ploughing the field.

- A man is walking on a rope.
- A man is walking across a rope.
- A man is balancing on a rope.
- A man is balancing on a rope at the beach.
- A man walks on a tightrope at the beach.
- A man is balancing on a volleyball net.
- A man is walking on a rope held by poles
- A man balanced on a wire.
- The man is balancing on the wire.
- A man is walking on a rope.
- A man is standing in the sea shore.

Venugopalan et al., “Translating Videos to Natural Language using Deep Recurrent Neural Networks”, NAACL-HTL 2015

Video Captioning

MT metrics (BLEU, METEOR) to compare the system generated sentences against (all) ground truth references.

Model	BLEU	METEOR
Best Prior Work [Thomason et al. COLING'14]	13.68	23.90
Only Images	12.66	20.96
Only Video	31.19	26.87
Images+Video	**33.29**	**29.07**

Pre-training only, no fine-tuning

No pre-training

Venugopalan et al., "Translating Videos to Natural Language using Deep Recurrent Neural Networks", NAACL-HTL 2015

Video Captioning

FGM: A person is dancing with the person on the stage.
YT: A group of men are riding the forest.
I+V: **A group of people are dancing.**
GT: Many men and women are dancing in the street.

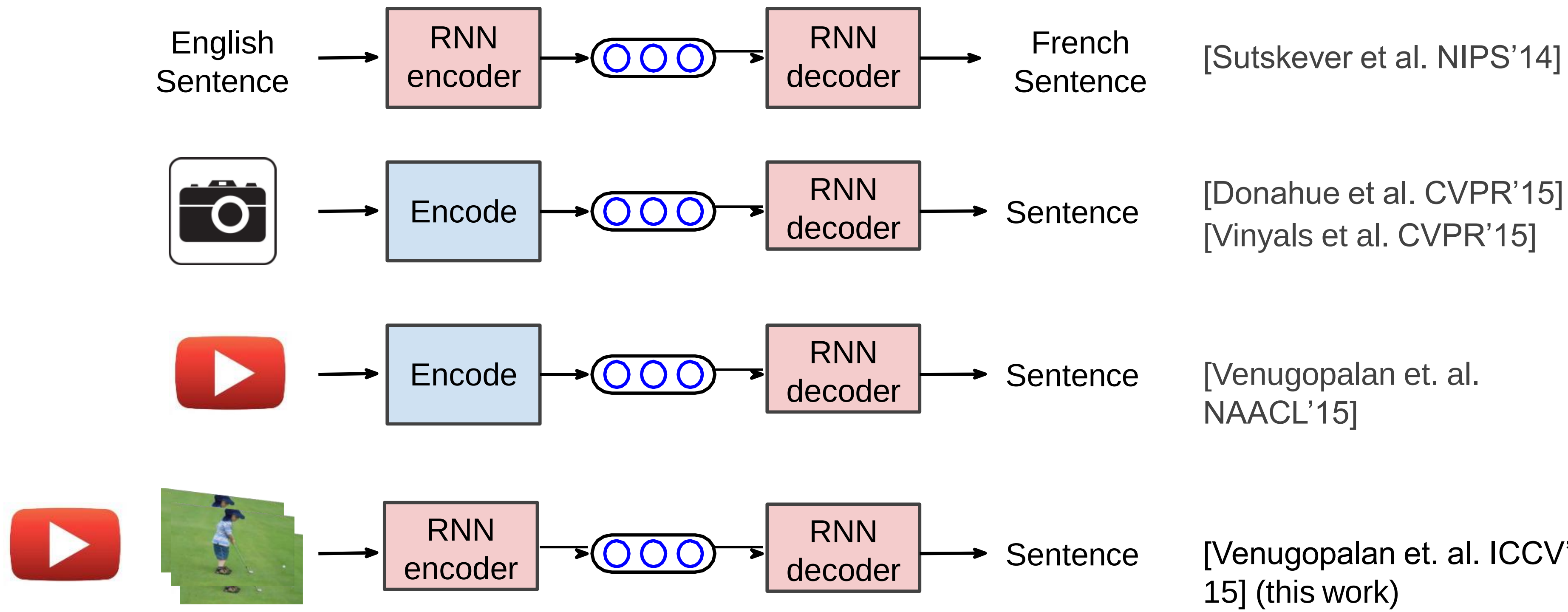
FGM: A person is cutting a potato in the kitchen.
YT: A man is slicing a tomato.
I+V: **A man is slicing a carrot.**
GT: A man is slicing carrots.

FGM: A person is walking with a person in the forest.
YT: A monkey is walking.
I+V: **A bear is eating a tree.**
GT: Two bear cubs are digging into dirt and plant matter at the base of a tree.

FGM: A person is riding a horse on the stage.
YT: A group of playing are playing in the ball.
I+V: **A basketball player is playing.**
GT: Dwayne wade does a fancy layup in an allstar game.

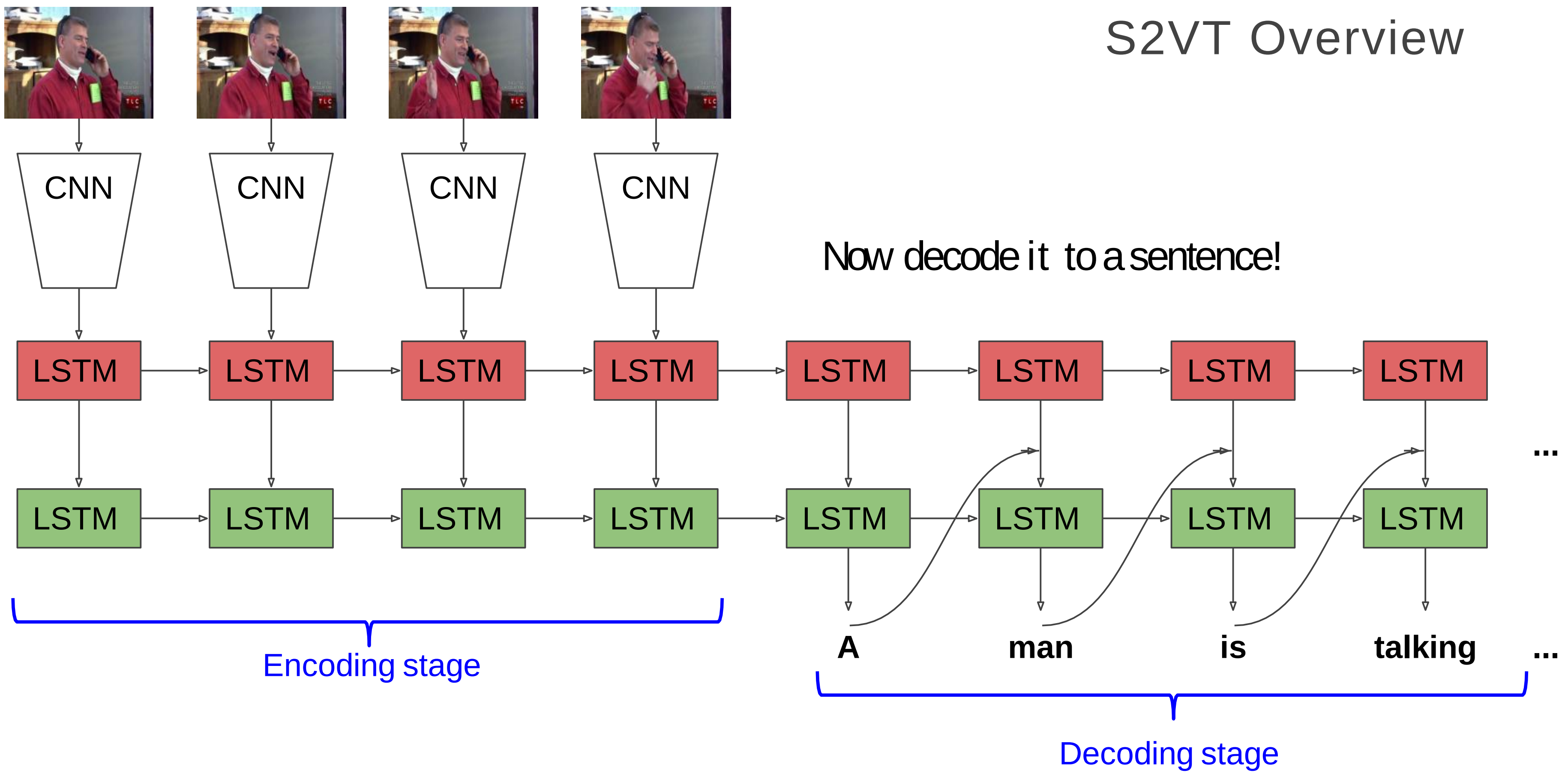
Venugopalan et al., “Translating Videos to Natural Language using Deep Recurrent Neural Networks”, NAACL-HTL 2015

Video Captioning



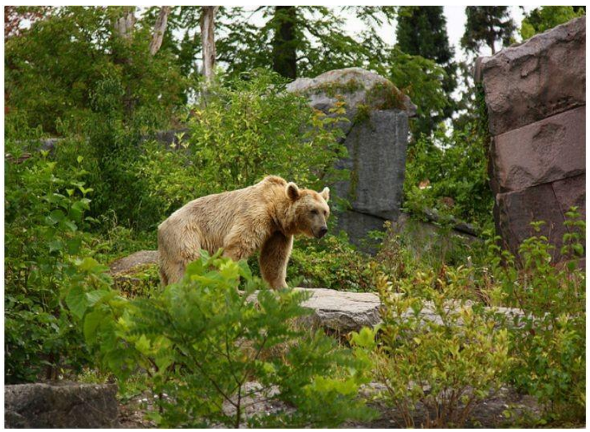
Venugopalan et al., "Sequence to Sequence - Video to Text", ICCV 2015

Video Captioning



Venugopalan et al., “Sequence to Sequence - Video to Text”, ICCV 2015

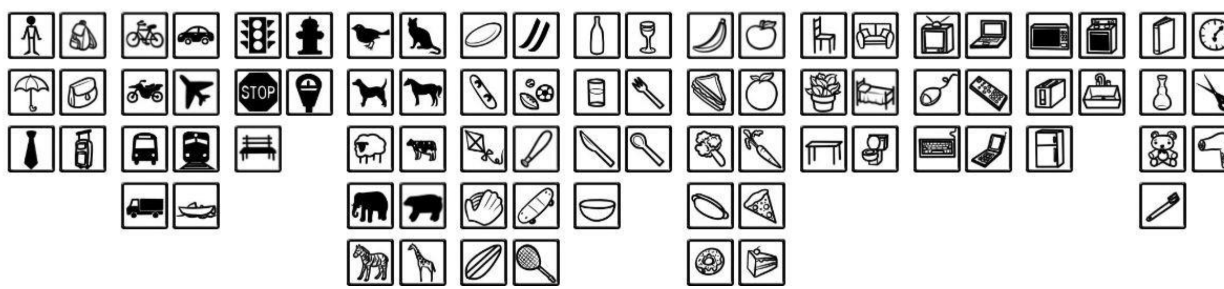
Visual Description



Berkeley LRCN [Donahue et al. CVPR'15]:
A brown bear standing on top of a lush green field.

MSR CaptionBot [http://captionbot.ai/]:
A large brown bear walking through a forest.

MSCOCO
80 classes

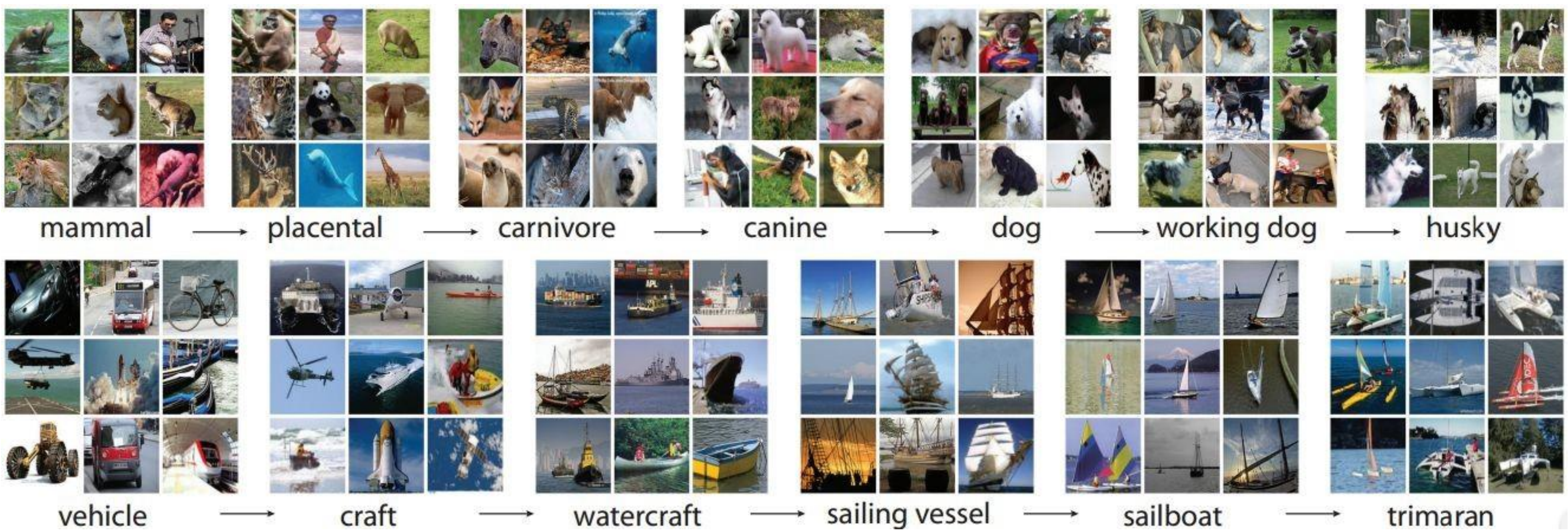


Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Object Recognition

Can identify hundreds of categories of objects.

IMAGENET 14M images, 22K classes [Deng et al. CVPR'09]



Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Novel Object Captioner (NOC)

We present Novel Object Captioner which can compose descriptions of 100s of objects in context.

NOC (ours): Describe novel objects <u>without</u> paired image-caption data.

IMAGENET + MSCOCO +

→ An **okapi** standing in the middle of a field.

Visual Classifiers. IMAGENET

→ **okapi**

Existing captioners. IMAGENET init + train MSCOCO

→ A horse standing in the dirt.

Venugopalan et al., "<u>Captioning Images With Diverse Objects</u>", CVPR 2017

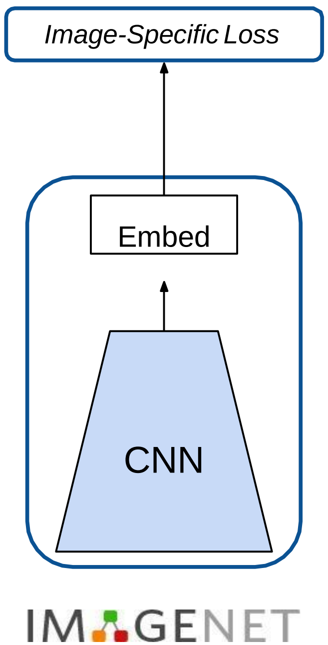
Insights

1. Need to recognize and describe objects outside of image-caption datasets.

okapi

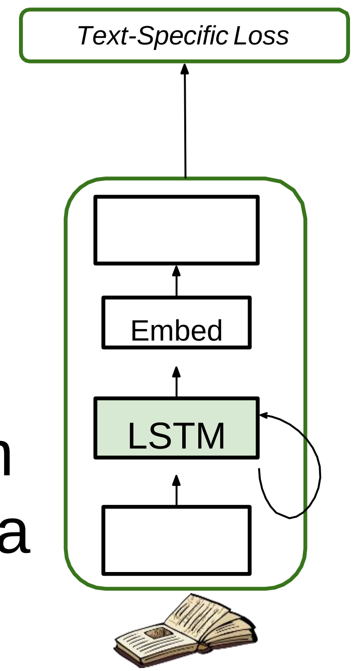
Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Insight 1: Train effectively on external sources



Visual features from unpaired image data

Language model from unannotated text data



Venugopalan et al., "[Captioning Images With Diverse Objects]", CVPR 2017

Insights

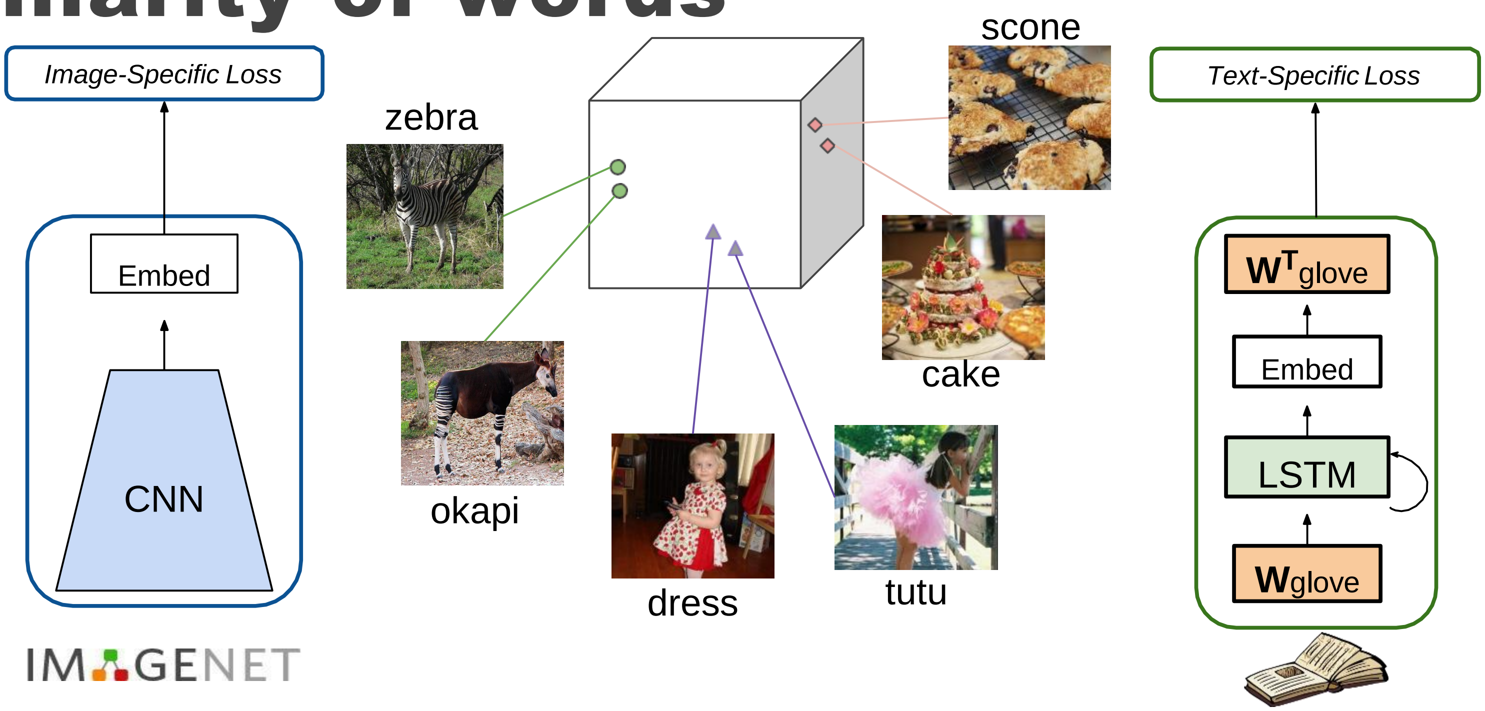
2. Describe unseen objects that are similar to objects seen in image-caption datasets.

okapi

zebra

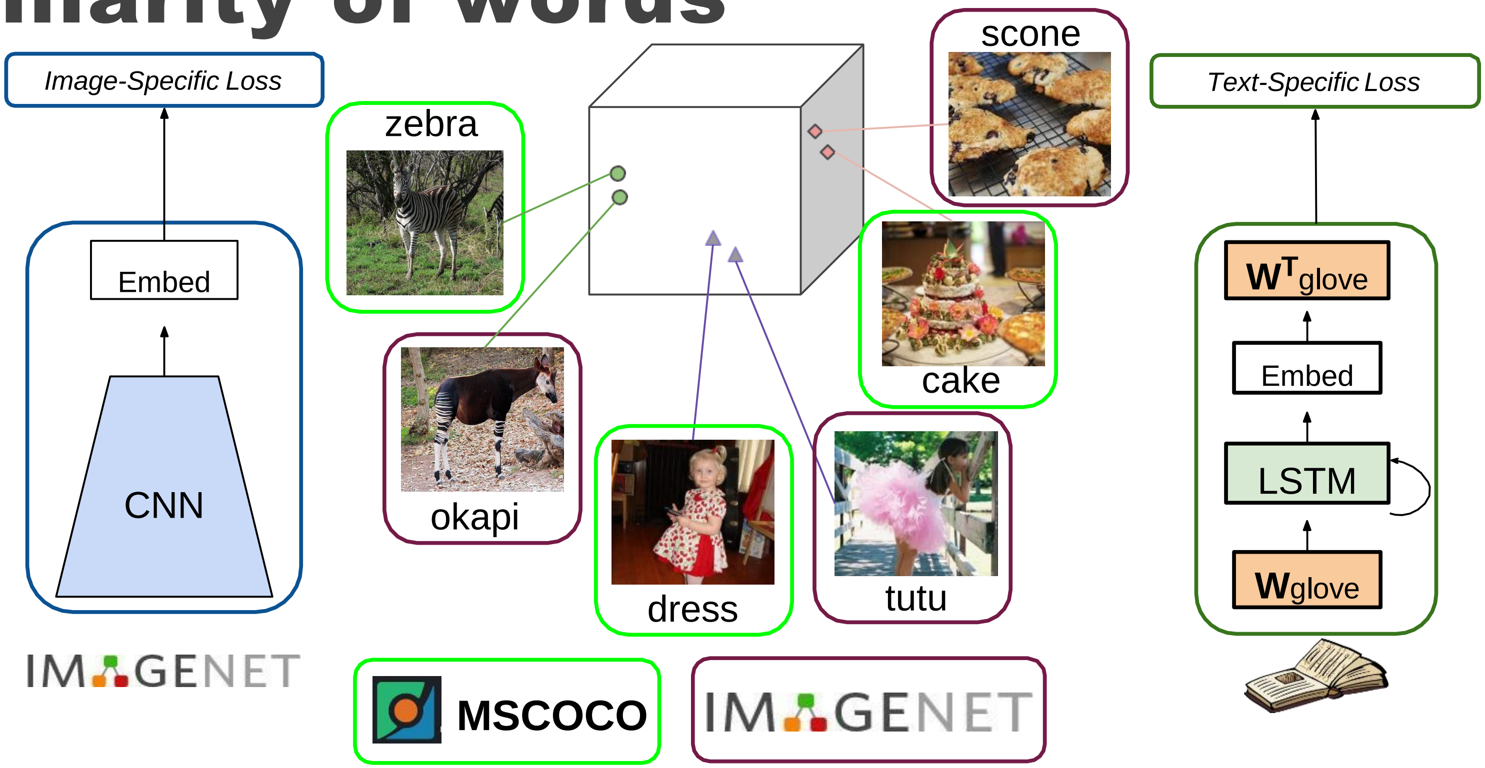
Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Insight 2: Capture semantic similarity of words



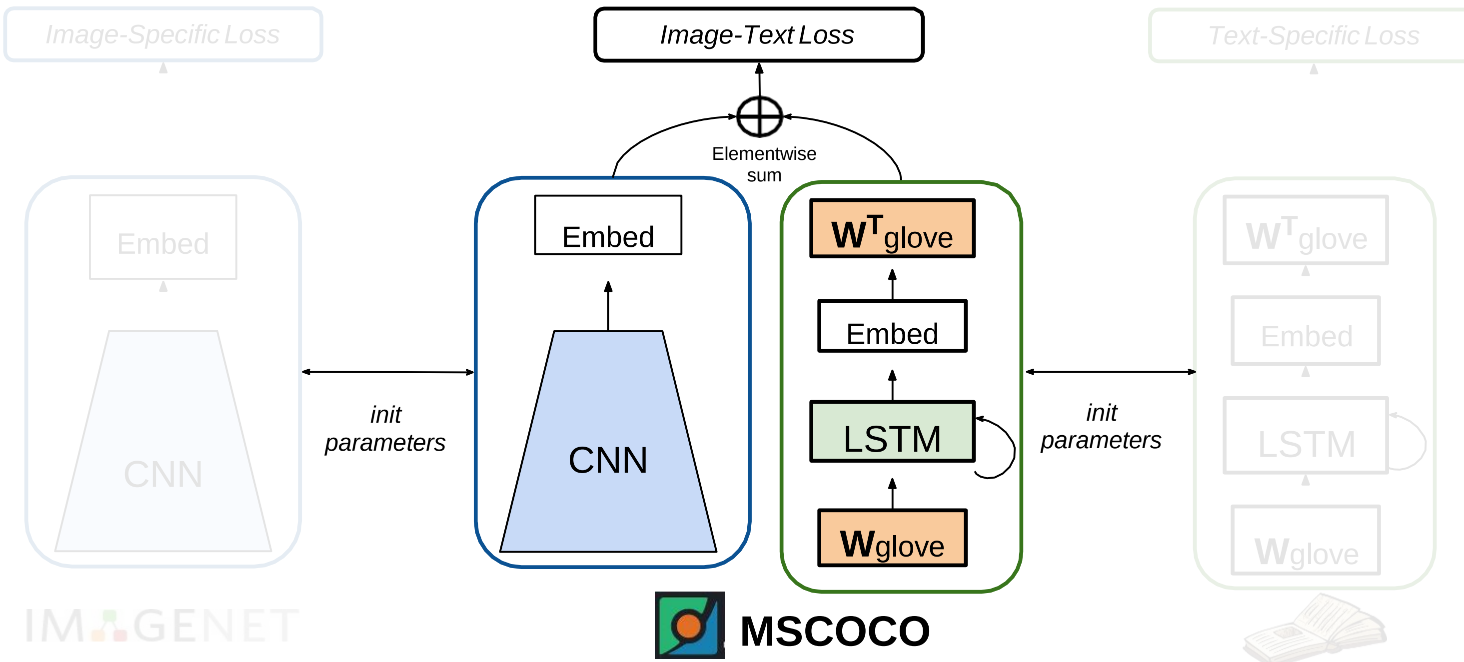
Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Insight 2: Capture semantic similarity of words



Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Combine to form a Caption Model

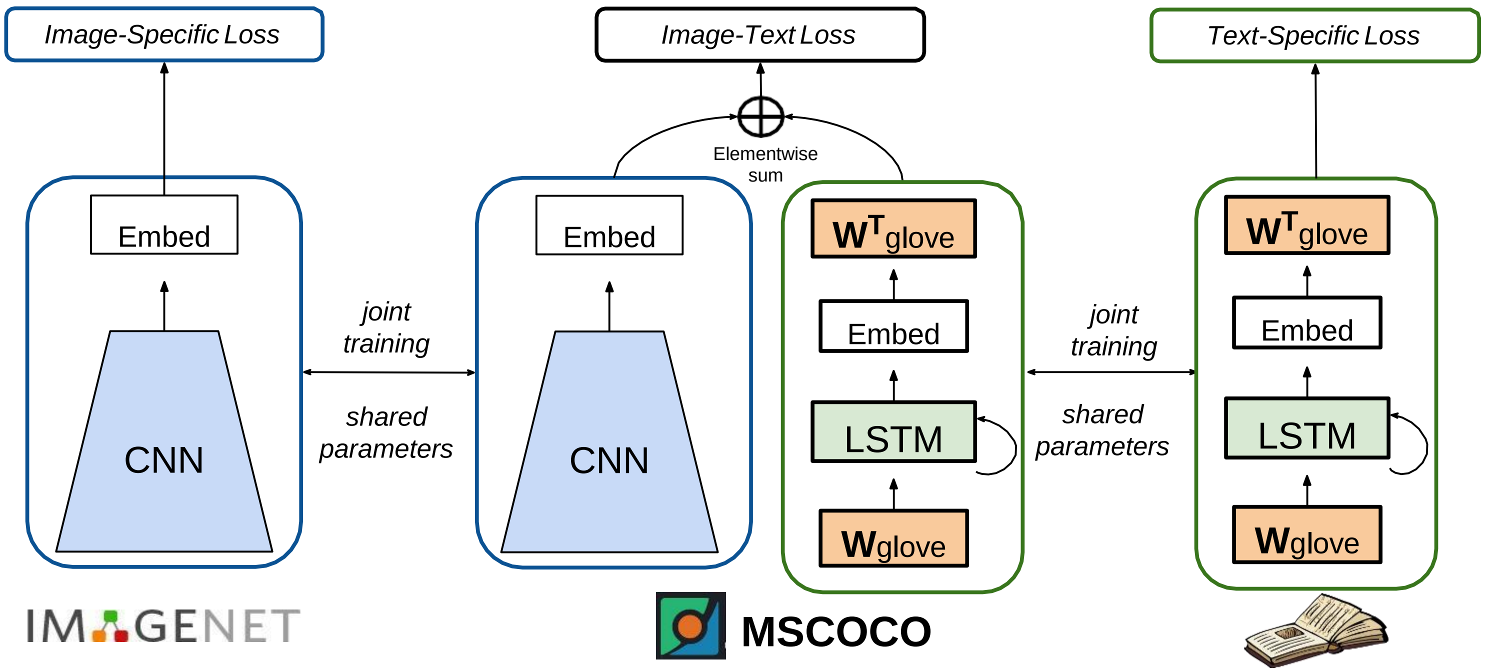


Not different from existing caption models. Problem: Forgetting.

[Catastrophic Forgetting in Neural Networks. Kirkpatrick et al. PNAS 2017]

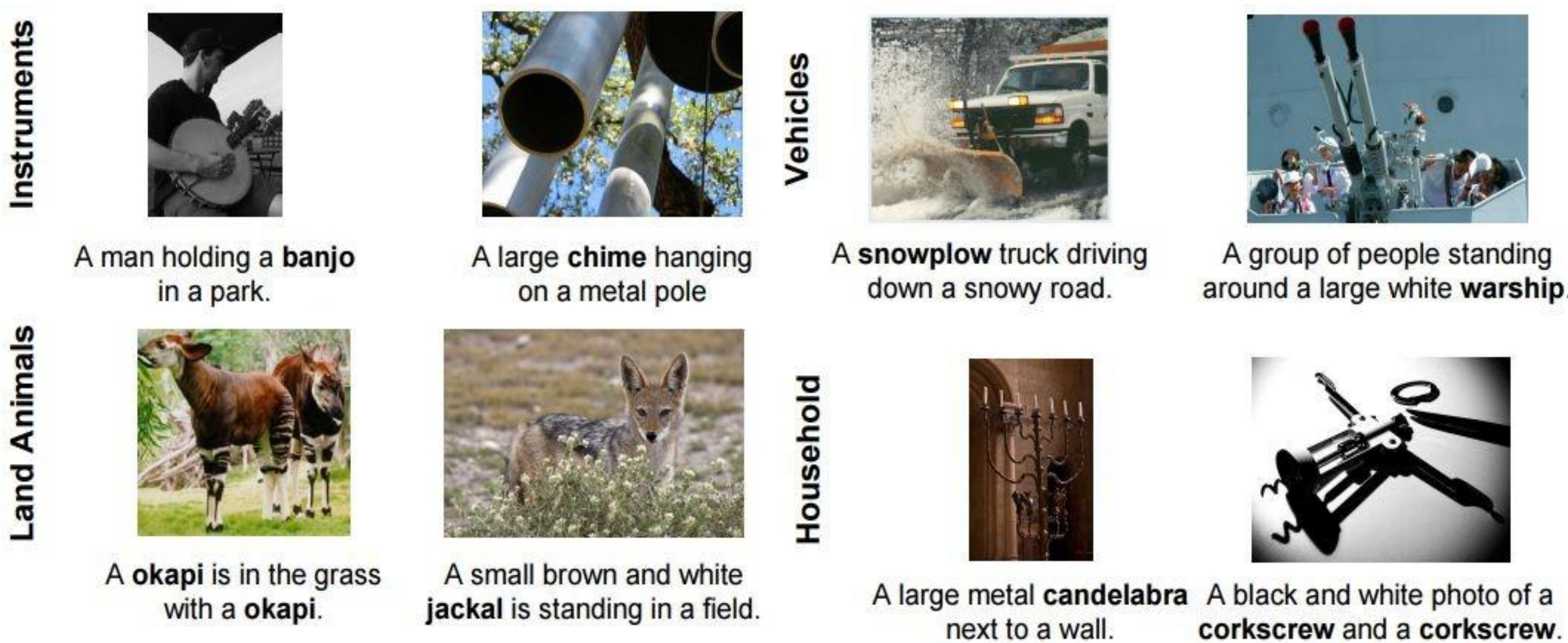
Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Insight 3: Jointly train on multiple sources



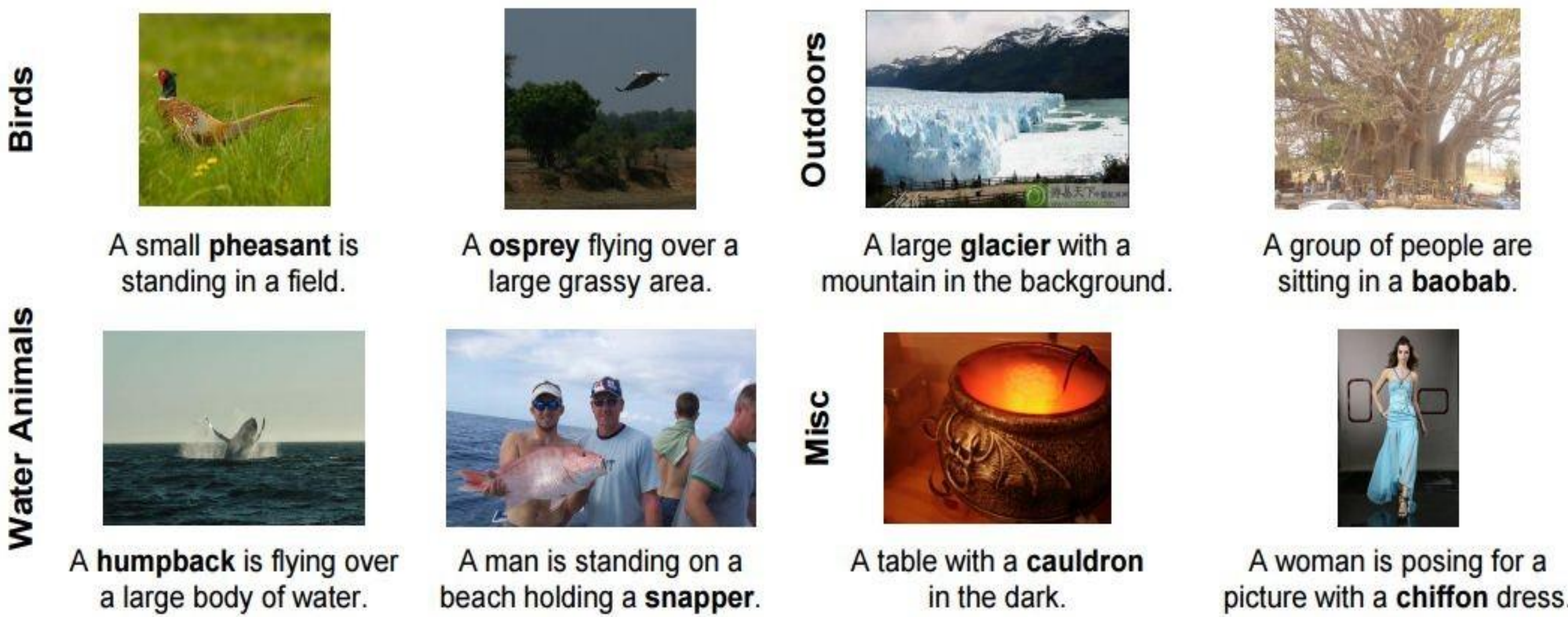
Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Qualitative Evaluation: ImageNet



Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Qualitative Evaluation: ImageNet

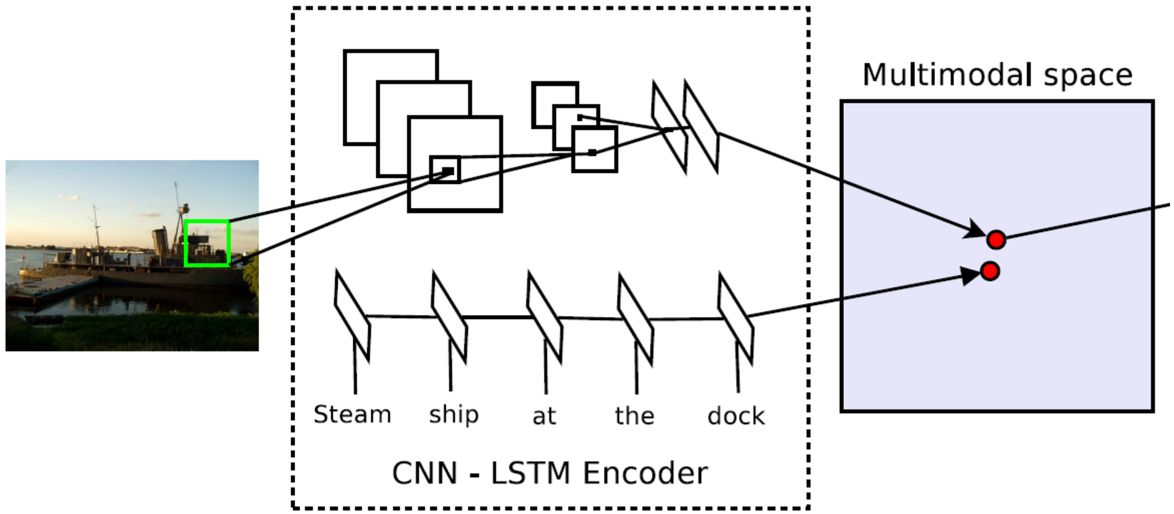


Venugopalan et al., "Captioning Images With Diverse Objects", CVPR 2017

Plan for this lecture

- Image captioning
  - Tool: Recurrent neural networks
  - Captioning for video
  - Diversifying captions
- Visual-semantic spaces
- Visual question answering
  - Incorporating knowledge and reasoning
  - Tool: Graph convolutional networks

Visual-semantic space

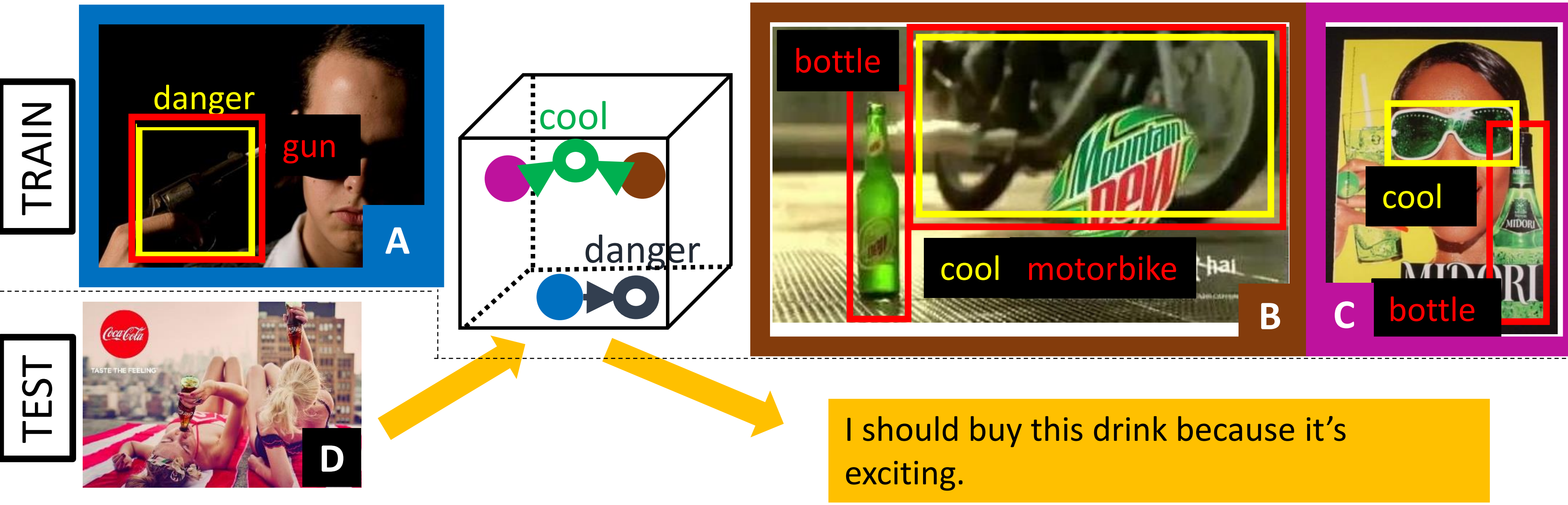


$$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

a denotes anchor
p denotes positive
n denotes negative

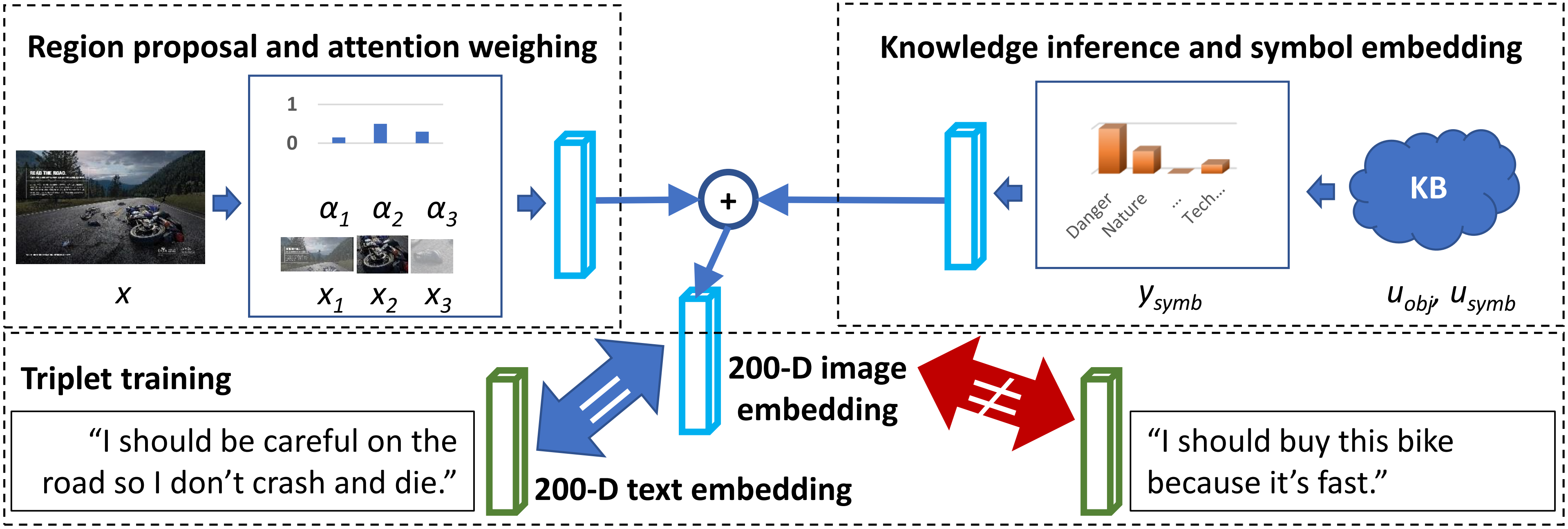
Kiros et al., “Unifying visual-semantic embeddings with multimodal neural language models”, TACL 2015

Visual-semantic space for understanding ads



Ye and Kovashka, "ADVISE: Symbolism and External Knowledge for Decoding Advertisements", ECCV 2018

Visual-semantic space for understanding ads



Ye and Kovashka, “ADVISE: Symbolism and External Knowledge for Decoding Advertisements”, ECCV 2018

Visual-semantic space for understanding ads

VSE++: "I should try this makeup because its fun."

Hussain-ranking: "I should stop smoking because it destroys your looks."

ADVISE (ours): "I should be careful to how I treat Earth because when the water leaves we die."

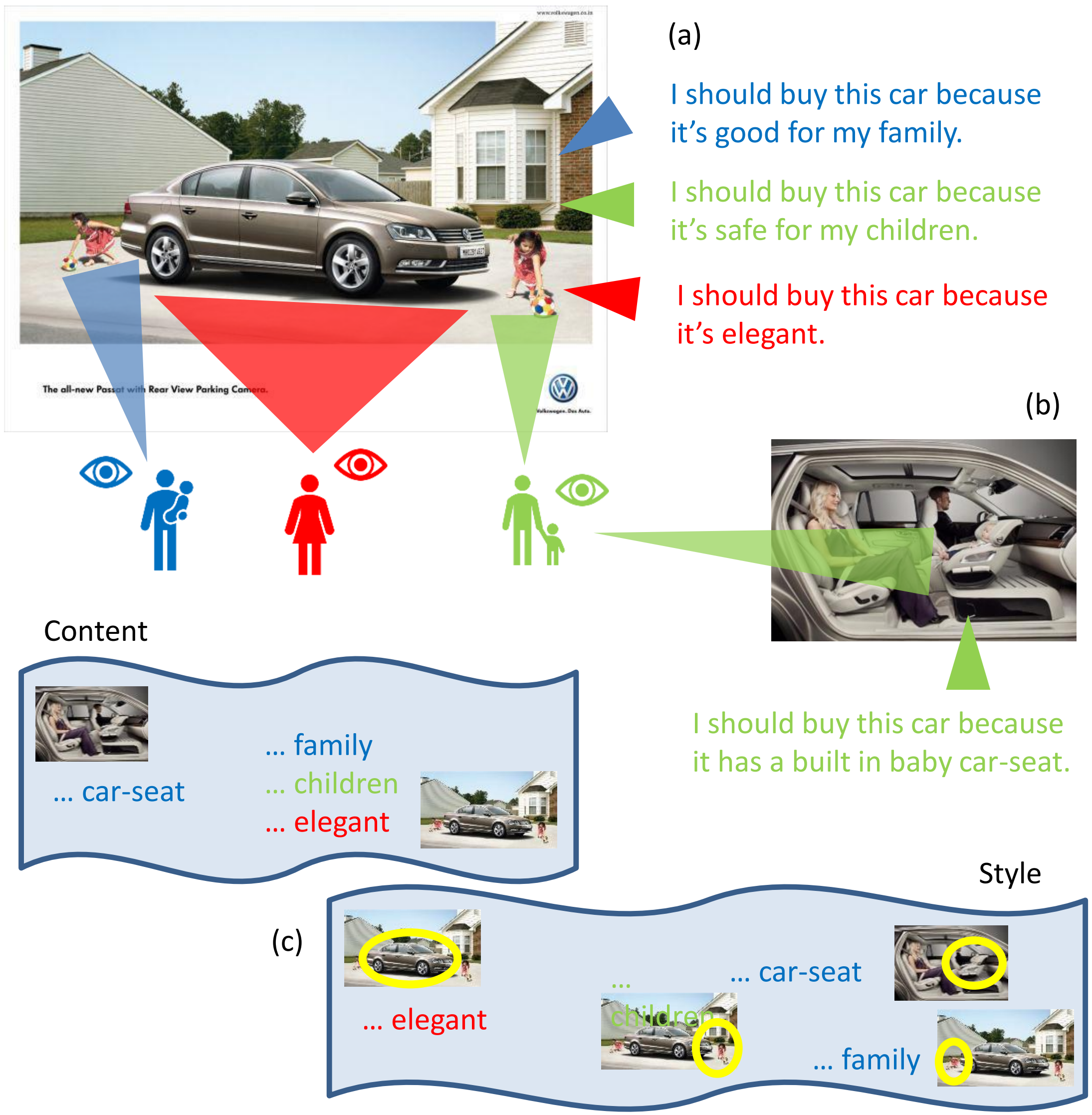
VSE++: "I should wear Nivea because it leaves no traces."

Hussain-ranking: "I should be eating these because it has fresh ingredients."

ADVISE (ours): "I should buy GeoPack paper because the their cutlery is eco-friendly."

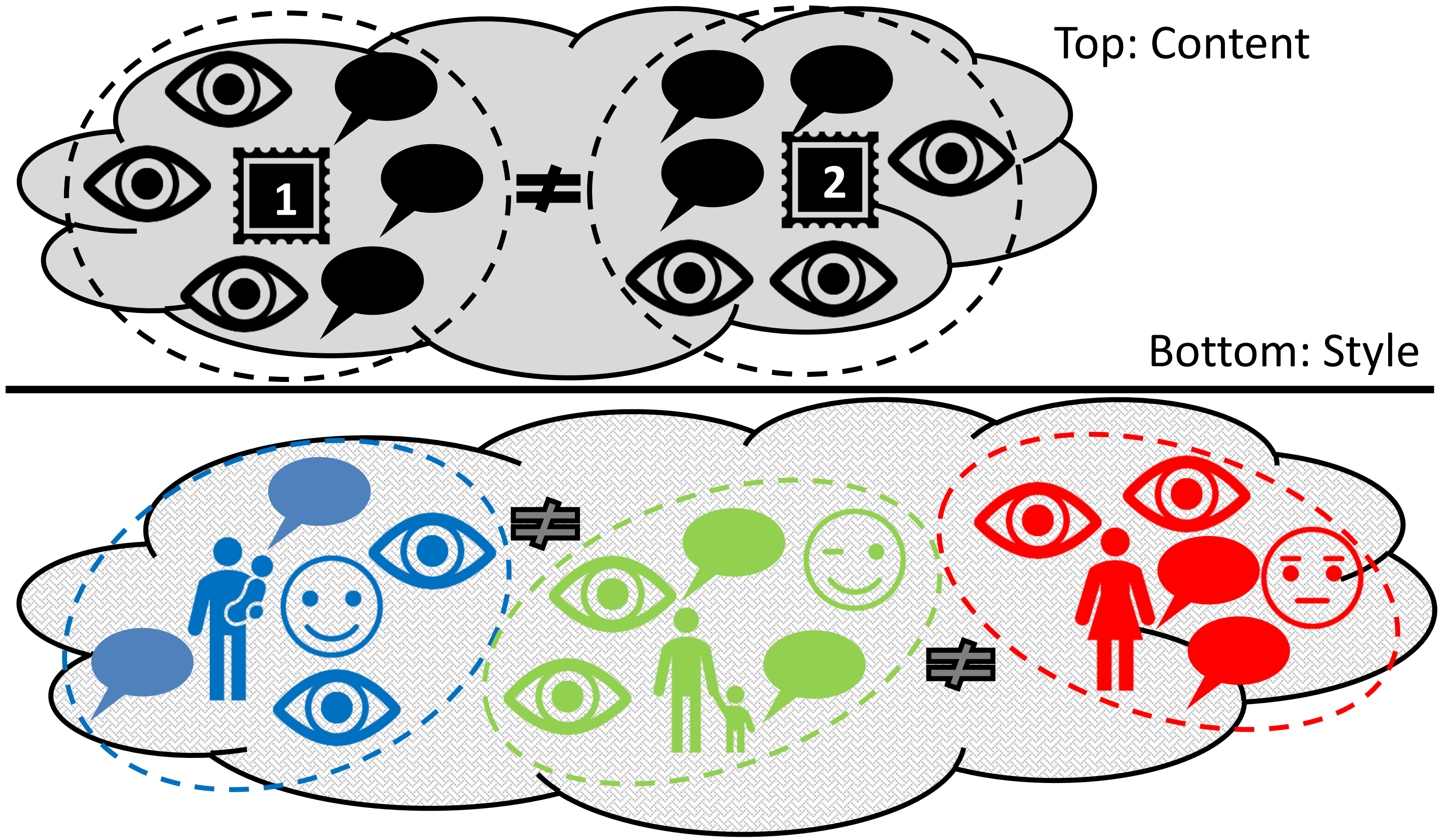
Ye and Kovashka, "ADVISE: Symbolism and External Knowledge for Decoding Advertisements", ECCV 2018

Style-aware visual-semantic spaces



Murrugarra-Llerena and Kovashka, "Cross-Modality Personalization for Retrieval", CVPR 2019

Style-aware visual-semantic spaces



Murrugarra-Llerena and Kovashka, “Cross-Modality Personalization for Retrieval”, CVPR 2019

Style-aware visual-semantic spaces

Content network

$$
\begin{aligned}
L_c(x, y, v; \theta) = \sum_{i=1}^{K} \Big[& \sum_{j \in N} \left[\|x_i^* - y_i^*\|_2^2 - \|x_i^* - y_j^*\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|y_i^* - x_i^*\|_2^2 - \|y_i^* - x_j^*\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|x_i^* - x_i^*\|_2^2 - \|x_i^* - x_j^*\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|y_i^* - y_i^*\|_2^2 - \|y_i^* - y_j^*\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|v_i^* - x_i^*\|_2^2 - \|v_i^* - x_j^*\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|v_i^* - y_i^*\|_2^2 - \|v_i^* - y_j^*\|_2^2 + \alpha\right]_+ \Big]
\end{aligned} \tag{1}
$$

Style network

$$
\begin{aligned}
L_s(x, y; \theta) = \sum_{i=1}^{K} \Big[& \sum_{j \in N} \left[\|x_*^a - y_*^a\|_2^2 - \|x_*^a - y_*^b\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|y_*^a - x_*^a\|_2^2 - \|y_*^a - x_*^b\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|x_*^a - x_*^a\|_2^2 - \|x_*^a - x_*^b\|_2^2 + \alpha\right]_+ \\
& + \sum_{j \in N} \left[\|y_*^a - y_*^a\|_2^2 - \|y_*^a - y_*^b\|_2^2 + \alpha\right]_+ \Big]
\end{aligned} \tag{2}
$$

{x, y} = {gaze, caption} OR {gaze, personality} OR {caption, personality}

Murrugarra-Llerena and Kovashka, “Cross-Modality Personalization for Retrieval”, CVPR 2019

Style-aware visual-semantic spaces

	Veit [41]	Base [10]	Style	**Ours**
g2p	0.1935	0.4269	0.4213	**0.4491**
t2p	**0.5806**	0.5639	0.5482	0.5741
p2g	0.162	0.438	0.4426	**0.4676**
t2g	0.5824	0.588	0.5259	**0.6037**
g2t	0.5871	0.5768	0.5158	**0.5963**
p2t	**0.5713**	0.537	0.5222	0.5528

Table 4. Top-3 accuracy for joint setup (higher is better). Content doesn't apply because it does not consider personality.

Murrugarra-Llerena and Kovashka, "Cross-Modality Personalization for Retrieval", CVPR 2019

Style-aware visual-semantic spaces

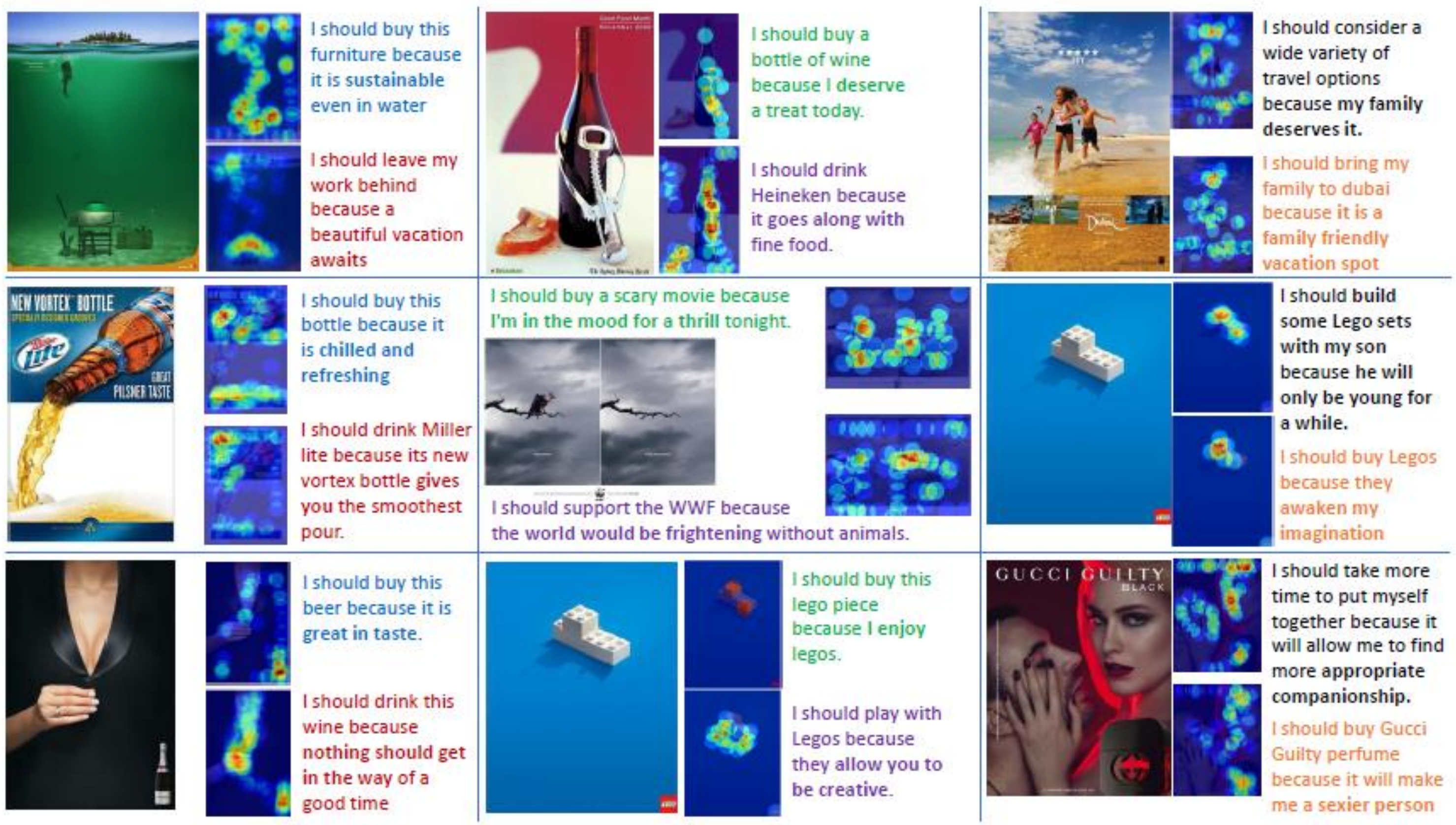


Figure 2. Text and gaze samples for different users. Each column represents a different set of images shown to two users per column. Gaze data is simulated via BubbleView interface [21], which produces data strongly correlated with gaze patterns.

Murrugarra-Llerena and Kovashka, "Cross-Modality Personalization for Retrieval", CVPR 2019

Plan for this lecture

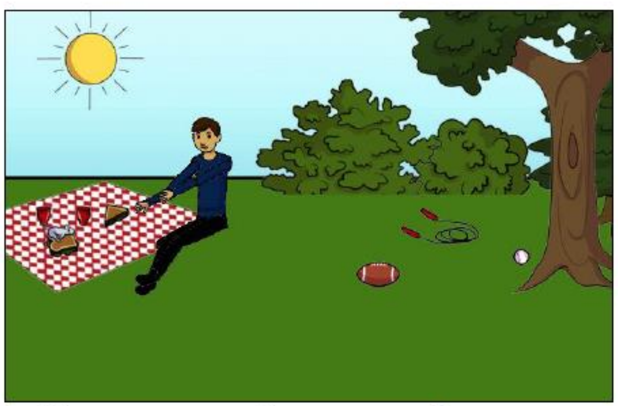
- Image captioning
  - Tool: Recurrent neural networks
  - Captioning for video
  - Diversifying captions
- Visual-semantic spaces
- Visual question answering
  - Incorporating knowledge and reasoning
  - Tool: Graph convolutional networks

Visual Question Answering (VQA)

Task: Given an image and a natural language open-ended question, generate a natural language answer.

What color are her eyes?
What is the mustache made of?

How many slices of pizza are there?
Is this a vegetarian pizza?

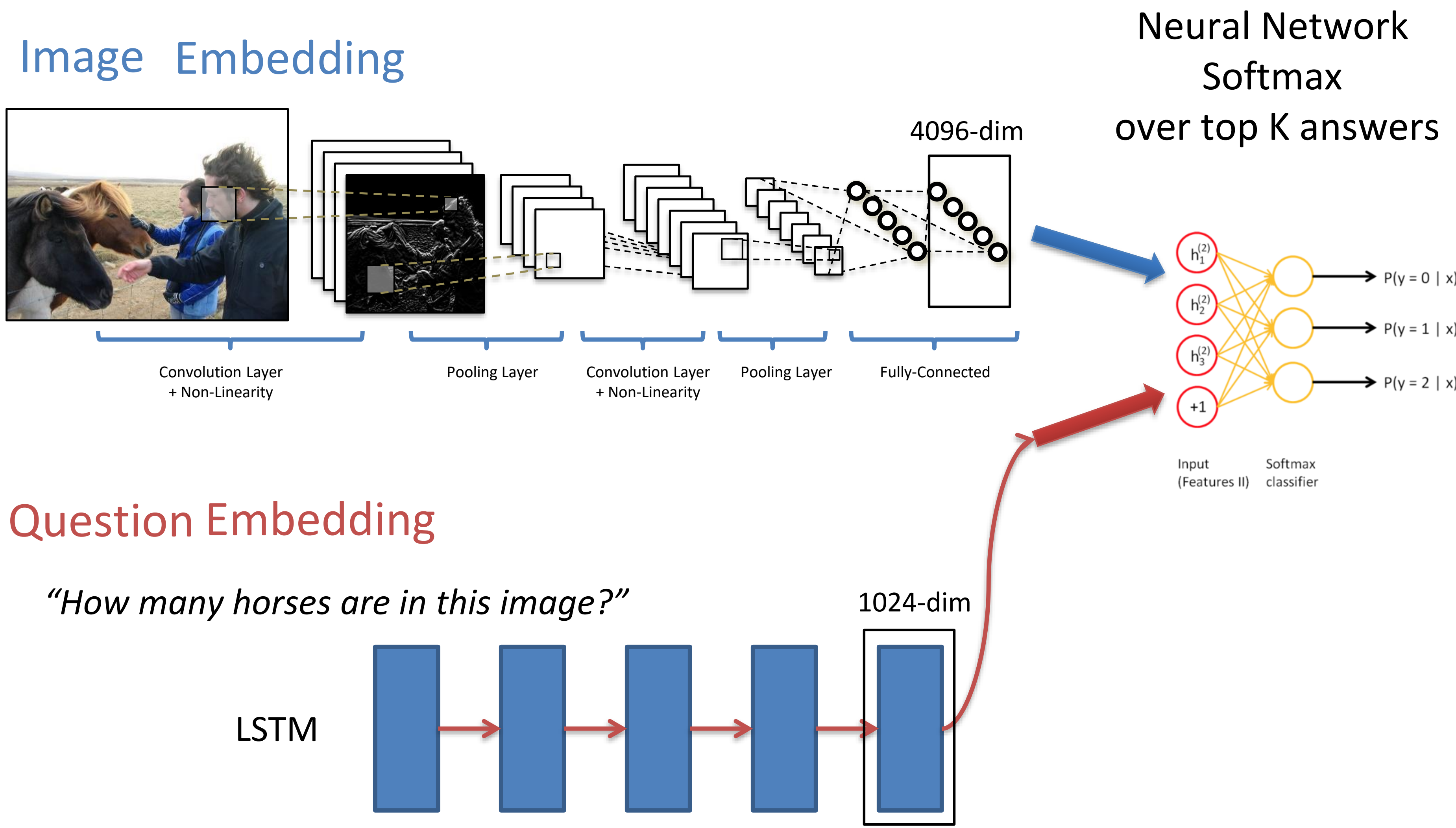


Is this person expecting company?
What is just under the tree?

Does it appear to be rainy?
Does this person have 20/20 vision?

Agrawal et al., "VQA: Visual Question Answering", ICCV 2015

Visual Question Answering (VQA)



Agrawal et al., "VQA: Visual Question Answering", ICCV 2015

Visual Question Answering (VQA)

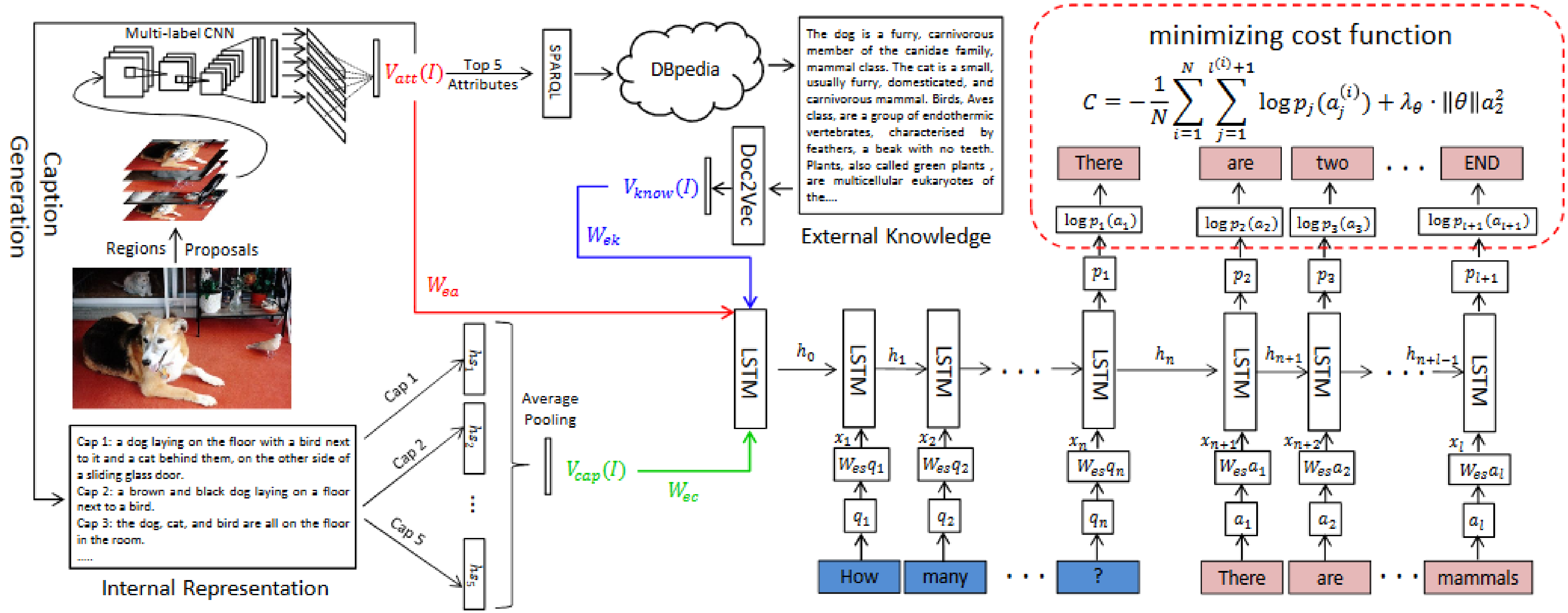
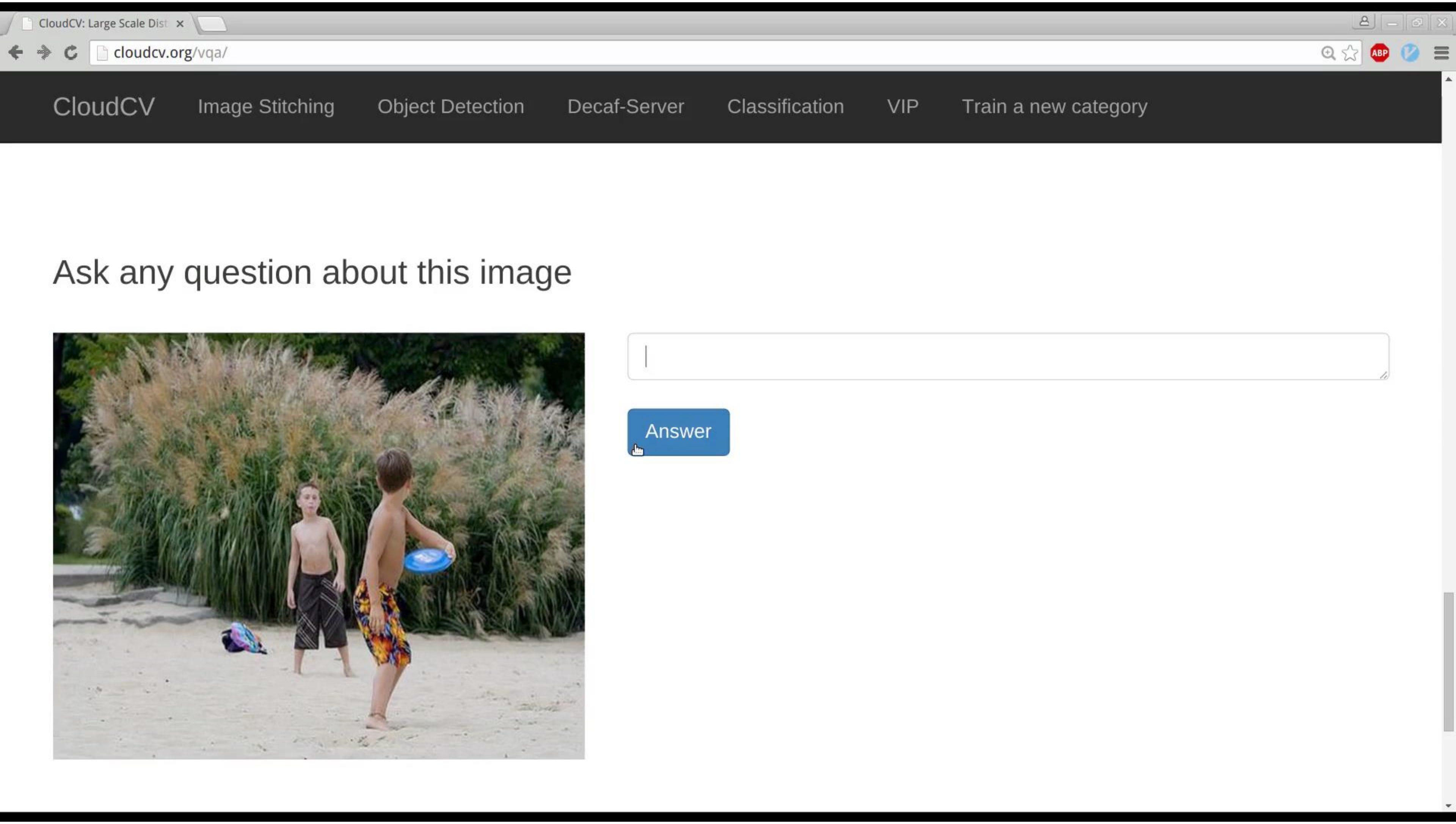


Figure 2. Our proposed framework: given an image, a CNN is first applied to produce the attribute-based representation $V_{att}(I)$. The internal textual representation is made up of image captions generated based on the image-attributes. The hidden state of the caption-LSTM after it has generated the last word in each caption is used as its vector representation. These vectors are then aggregated as $V_{cap}(I)$ with average-pooling. The external knowledge is mined from the KB (in this case DBpedia) and the responses encoded by Doc2Vec, which produces a vector $V_{know}(I)$. The 3 vectors V are combined into a single representation of scene content, which is input to the VQA LSTM model which interprets the question and generates an answer.

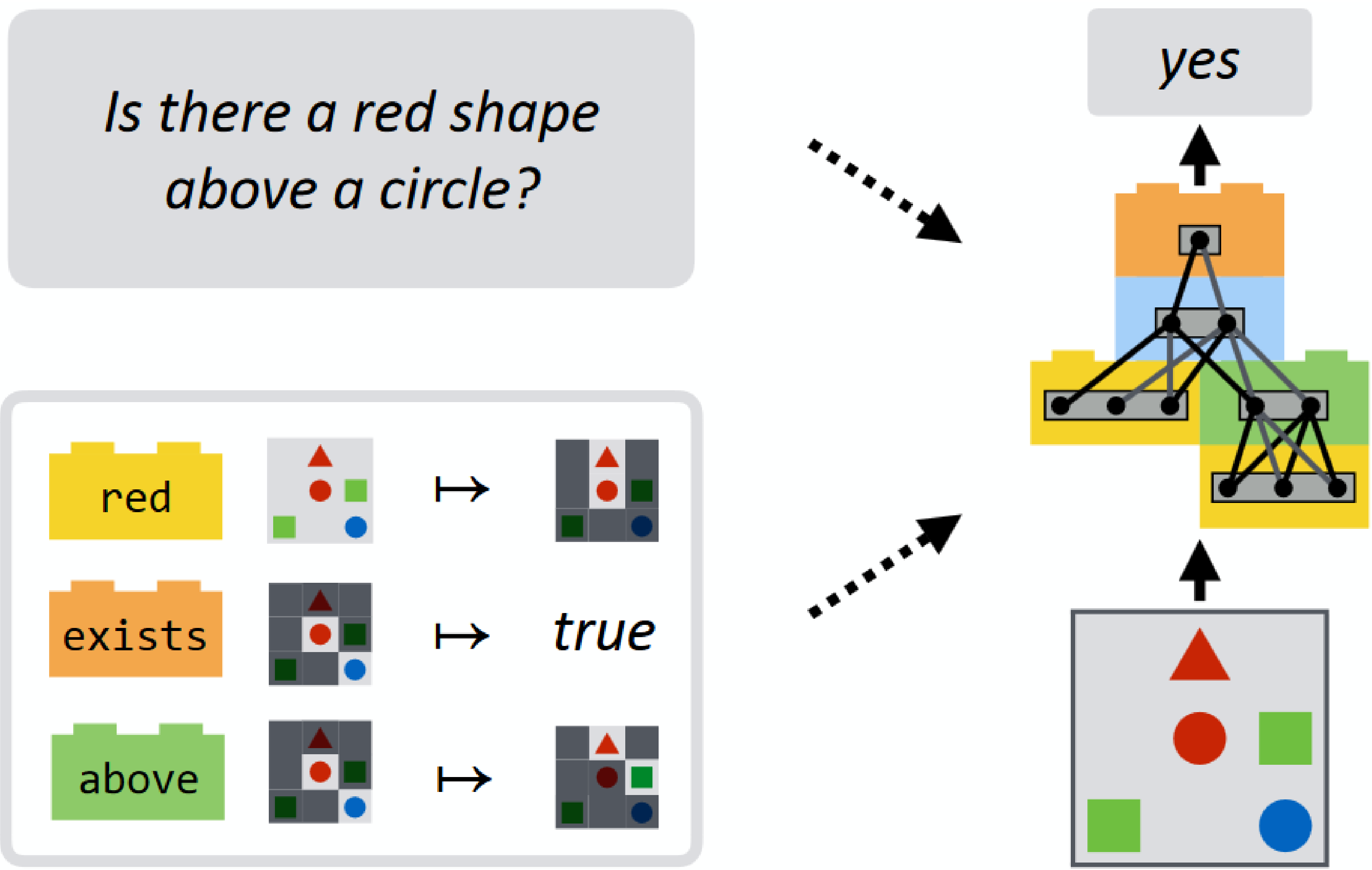
Wu et al., "Ask Me Anything: Free-Form Visual Question Answering Based on Knowledge From External Sources", CVPR 2016

Visual Question Answering (VQA)



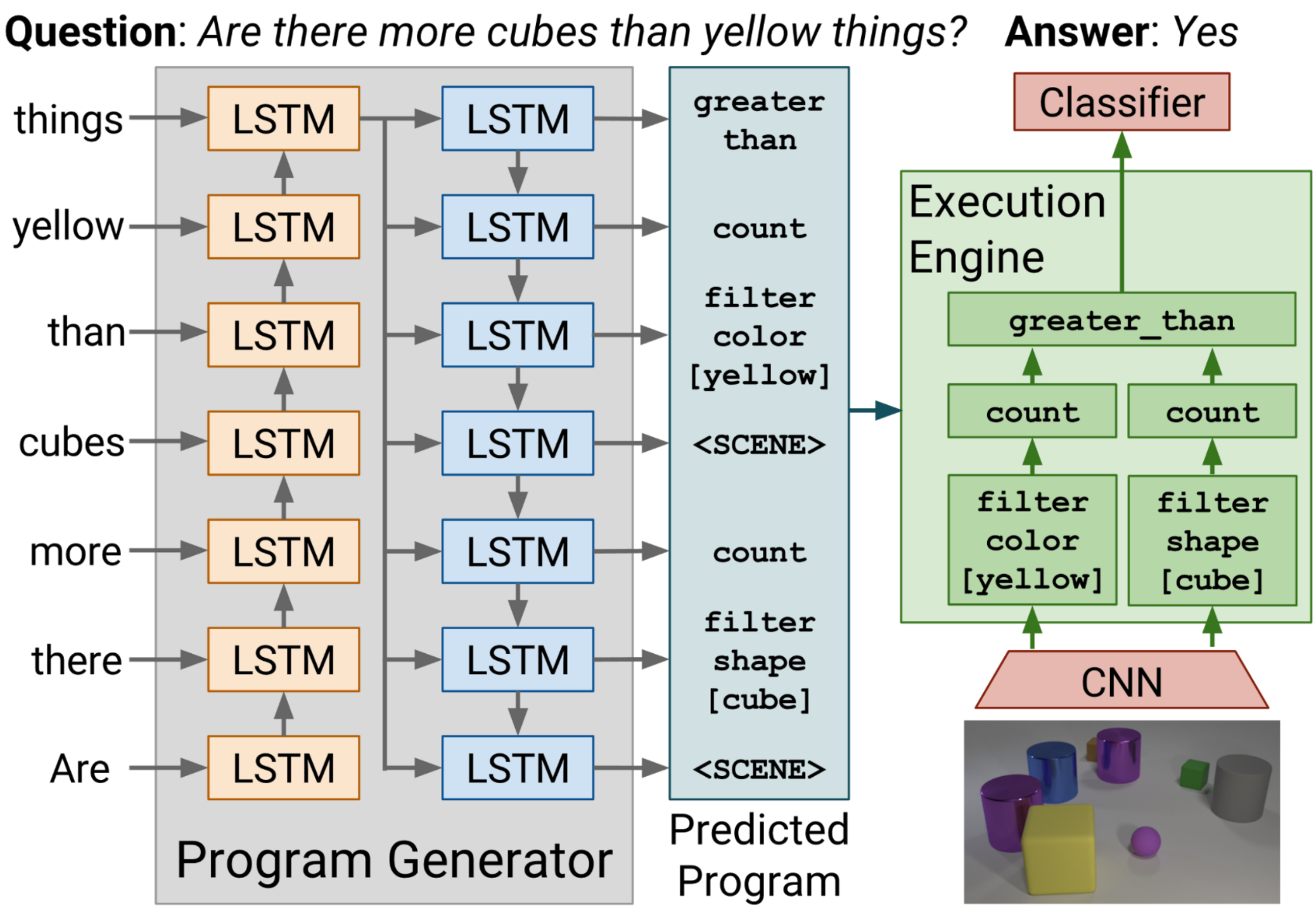
Agrawal et al., “VQA: Visual Question Answering”, ICCV 2015

Reasoning for VQA



Andreas et al., "[Neural Module Networks]", CVPR 2016

Reasoning for VQA



Johnson et al., "[Inferring and Executing Programs for Visual Reasoning]()", ICCV 2017

Reasoning for VQA

Question: Which object in the image can be used to eat with?
Relation: UsedFor
Associated Fact: (Fork, UsedFor, Eat)
Answer Source: Image
Answer: Fork

Question: What do the animals in the image eat?
Relation: RelatedTo
Associated Fact: (Sheep, RelatedTo, Grass Eater)
Answer Source: Knowledge Base
Answer: Grass

Question: Which equipment in this image is used to hit baseball?
Relation: CapableOf
Associated Fact: (Baseball bat, CapableOf, Hit a baseball)
Answer Source: Image
Answer: Baseball bat

Fig. 1. The FVQA dataset expects methods to answer questions about images utilizing information from the image, as well as fact-based knowledge bases. Our method makes use of the image, and question text features, as well as high-level visual concepts extracted from the image in combination with a learned fact-ranking neural network. Our method is able to answer both visually grounded as well as fact based questions.

Narasimhan and Schwing, “Straight to the Facts: Learning Knowledge Base Retrieval for Factual Visual Question Answering”, ECCV 2018

Reasoning for VQA

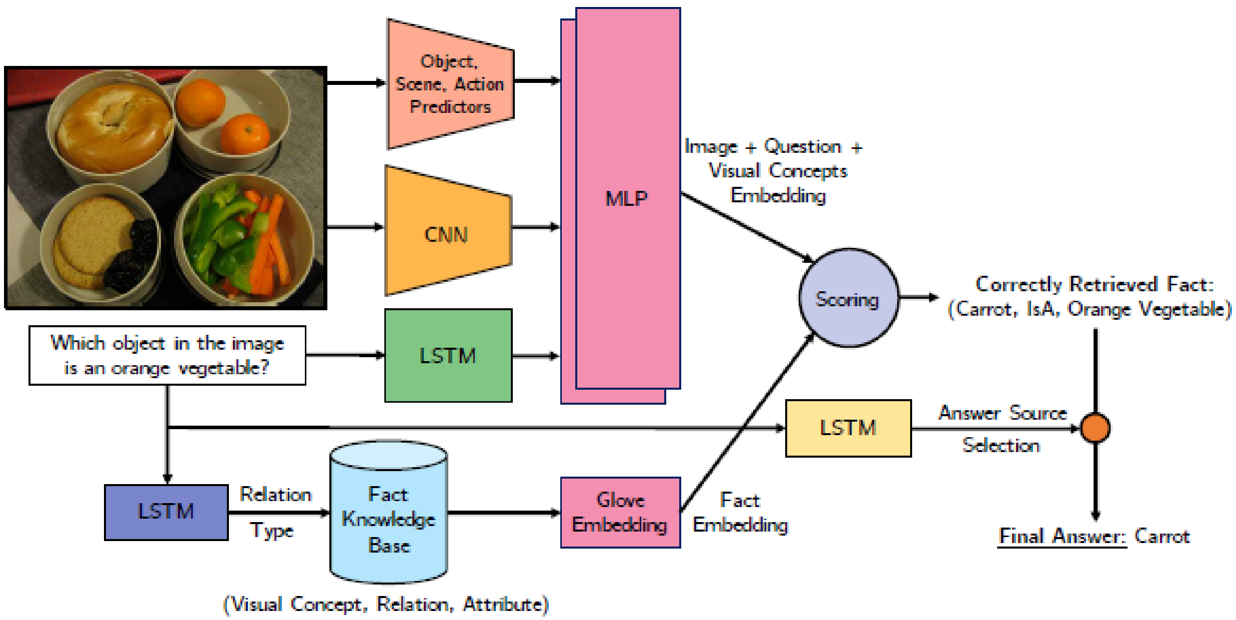


Fig. 2. Overview of the proposed approach. Given an image and a question about the image, we obtain an Image + Question Embedding through the use of a CNN on the image, an LSTM on the question, and a Multi Layer Perceptron (MLP) for combining the two modalities. In order to filter relevant facts from the Knowledge Base (KB), we use another LSTM to predict the fact relation type from the question. The retrieved structured facts are encoded using GloVe embeddings. The retrieved facts are ranked through a dot product between the embedding vectors and the top-ranked fact is returned to answer the question.

Narasimhan and Schwing, "Straight to the Facts: Learning Knowledge Base Retrieval for Factual Visual Question Answering", ECCV 2018

Reasoning for VQA

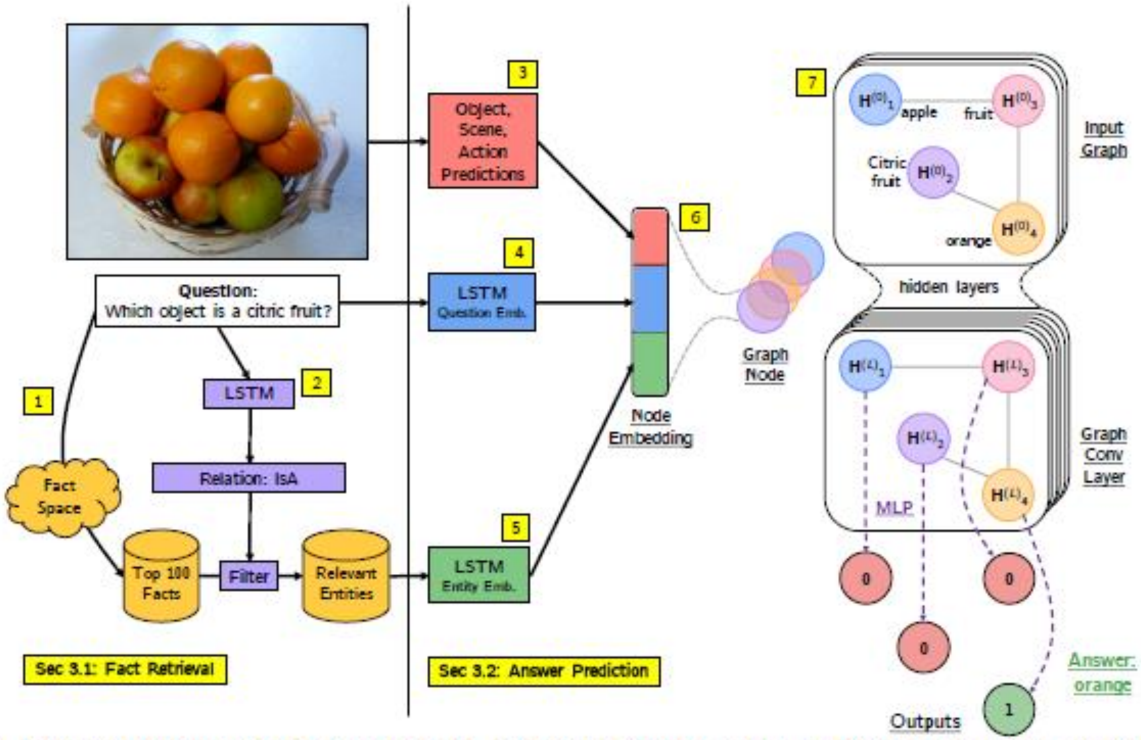
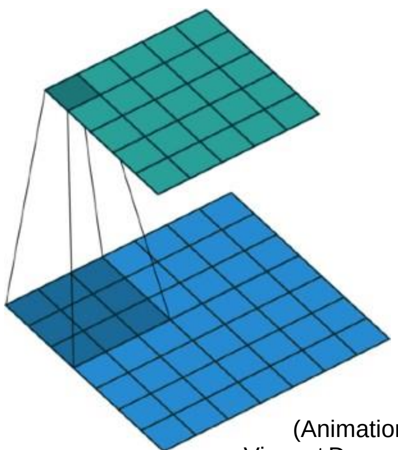


Figure 2: Outline of the proposed approach: Given an image and a question, we use a similarity scoring technique (1) to obtain relevant facts from the fact space. An LSTM (2) predicts the relation from the question to further reduce the set of relevant facts and its entities. An entity embedding is obtained by concatenating the visual concepts embedding of the image (3), the LSTM embedding of the question (4), and the LSTM embedding of the entity (5). Each entity forms a single node in the graph and the relations constitute the edges (6). A GCN followed by an MLP performs joint assessment (7) to predict the answer. Our approach is trained end-to-end.

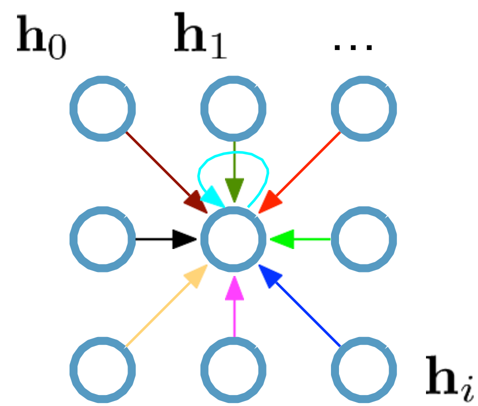
Narasimhan and Schwing, “Out of the Box: Reasoning with Graph Convolution Nets for Factual Visual Question Answering”, NeurIPS 2018

Graph convolutional networks

Recall: Single CNN layer with 3x3 filter:



(Animation by Vincent Dumoulin)



Update for a single pixel:

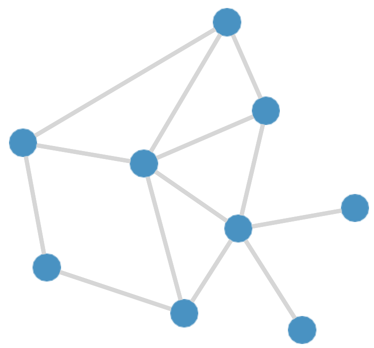
- Transform messages individually $\mathbf{W}_i \mathbf{h}_i$
- Add everything up $\sum_i \mathbf{W}_i \mathbf{h}_i$

Full update: $$\mathbf{h}_4^{(l+1)} = \sigma\left(\mathbf{W}_0^{(l)} \mathbf{h}_0^{(l)} + \mathbf{W}_1^{(l)} \mathbf{h}_1^{(l)} + \cdots + \mathbf{W}_8^{(l)} \mathbf{h}_8^{(l)}\right)$$

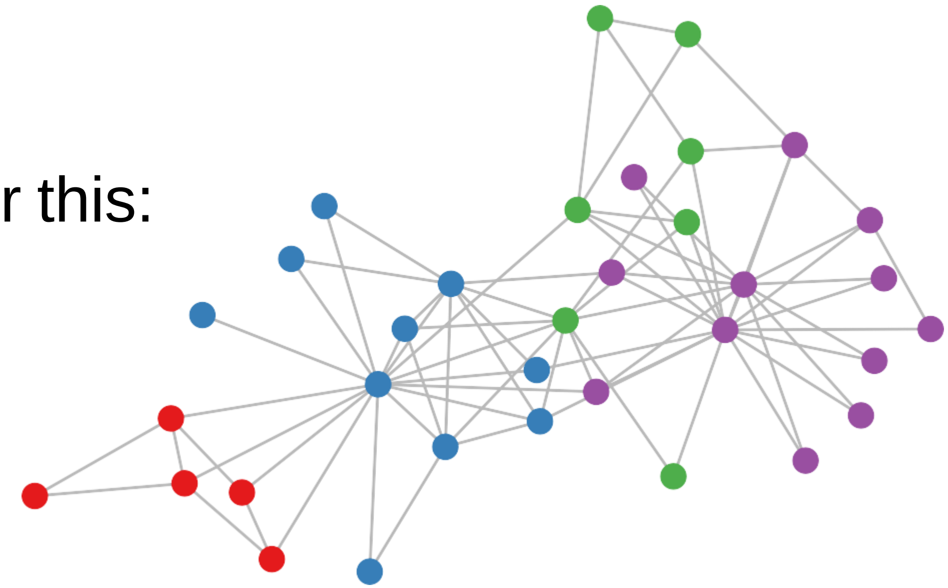
Kipf and Welling, “Semi-supervised learning with deep generative models”, ICLR 2017 (slides by Thomas Kipf)

Graph convolutional networks

What if our data looks like this?



or this:



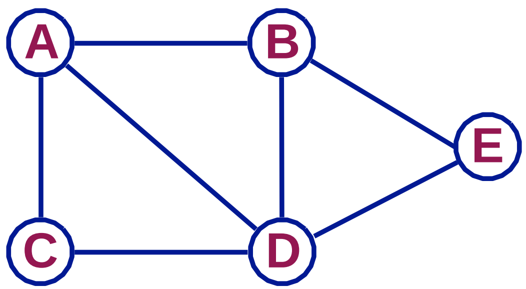
Real-world examples:

- Social networks
- World-wide-web
- Protein-interaction networks
- Telecommunication networks
- Knowledge graphs
- …

Kipf and Welling, "Semi-supervised learning with deep generative models", ICLR 2017 (slides by Thomas Kipf)

Graph convolutional networks

Graph: $G = (\mathcal{V}, \mathcal{E})$



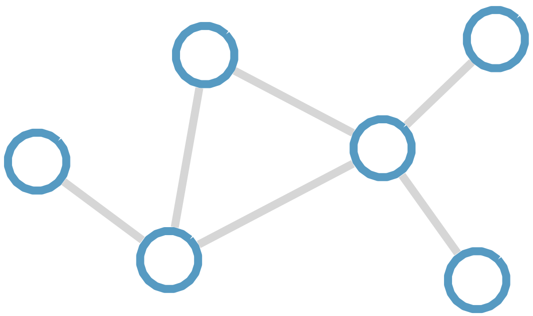
Adjacency matrix: A

$$\begin{array}{c|c} & \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \end{matrix} & \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

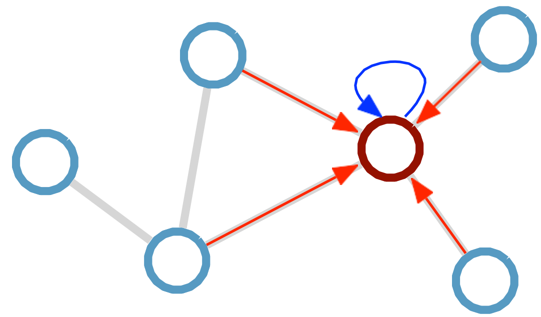
Kipf and Welling, "Semi-supervised learning with deep generative models", ICLR 2017 (slides by Thomas Kipf)

Graph convolutional networks

Consider this undirected graph:



Calculate update for node in red:



Update rule:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)} \mathbf{W}_0^{(l)} + \sum_{j \in \mathcal{N}_i} \frac{1}{c_{ij}} \mathbf{h}_j^{(l)} \mathbf{W}_1^{(l)} \right)$$

$\mathcal{N}_i$: neighbor indices

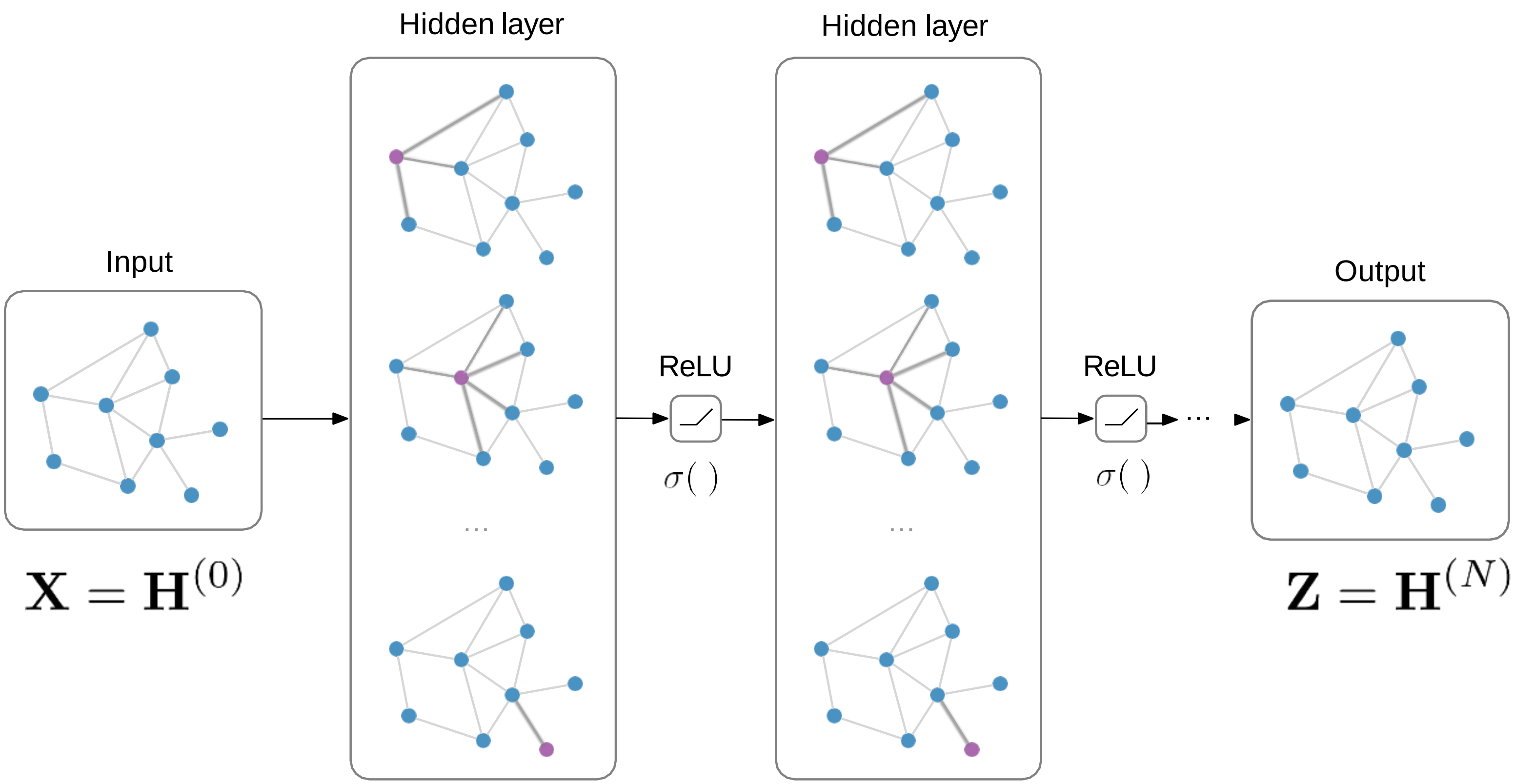
c_{ij} : norm. constant (per edge)

Note: We could also choose simpler or more general functions over the neighborhood

Kipf and Welling, “Semi-supervised learning with deep generative models”, ICLR 2017 (slides by Thomas Kipf)

Graph convolutional networks

Input: Feature matrix $\mathbf{X} \in \mathbb{R}^{N \times E}$, preprocessed adjacency matrix $\hat{\mathbf{A}}$



$$\mathbf{H}^{(l+1)} = \sigma\left(\hat{\mathbf{A}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}\right)$$

Kipf and Welling, “Semi-supervised learning with deep generative models”, ICLR 2017 (slides by Thomas Kipf)

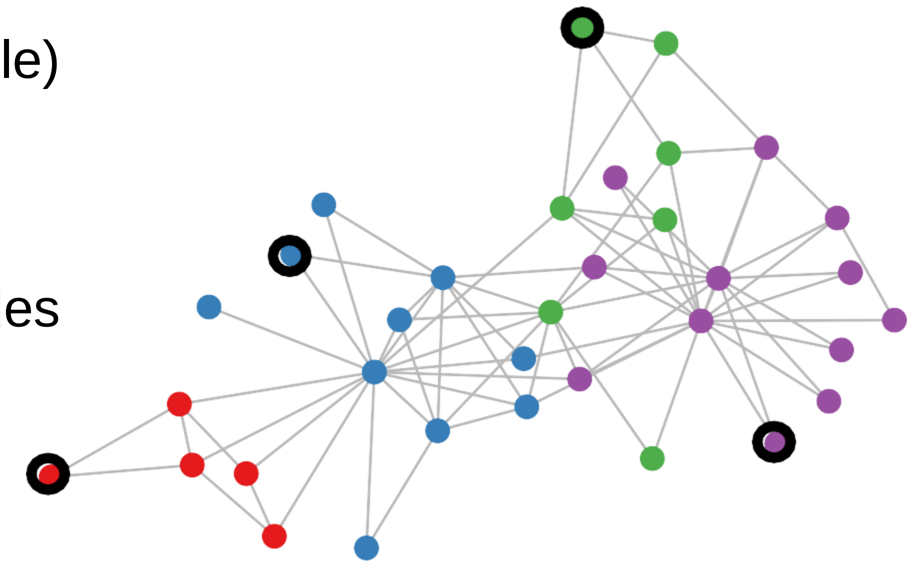
Graph convolutional networks

Semi-supervised classification on graphs

Setting:

Some nodes are labeled (black circle)
All other nodes are unlabeled

Task:
Predict node label of unlabeled nodes



Kipf and Welling, “Semi-supervised learning with deep generative models”, ICLR 2017 (slides by Thomas Kipf)

What’s next?