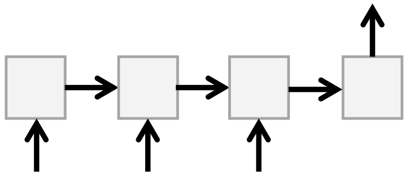
Sequence models

Multiple - Single



Sequence Classification

Single - Multiple

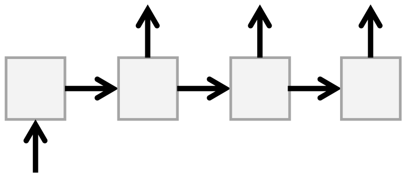


Image Captioning

Multiple - Multiple

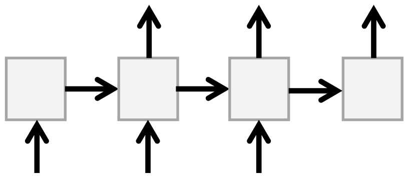
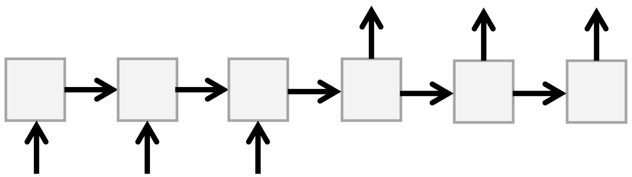


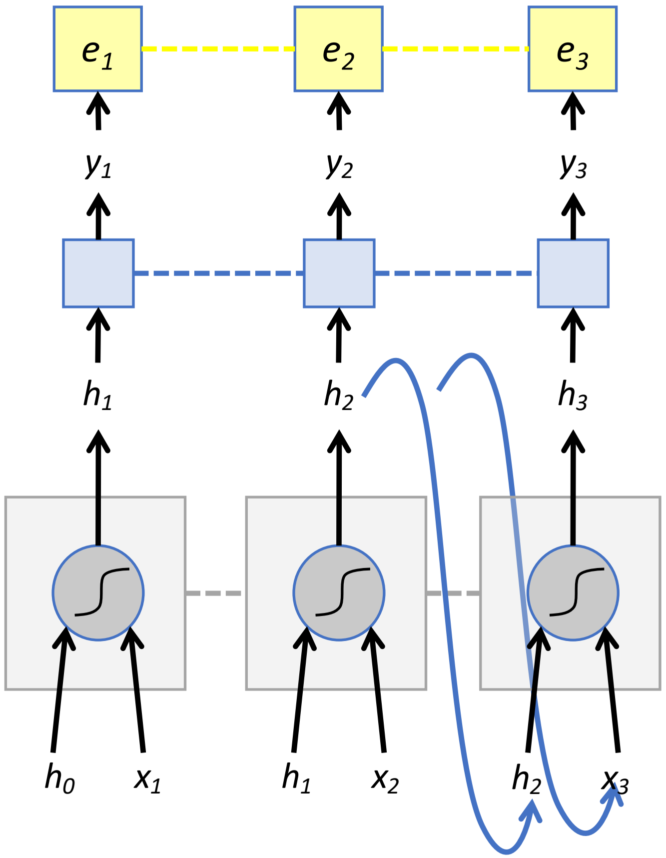
Image Captioning

Multiple - Multiple



Translation

RNN Forward Pass



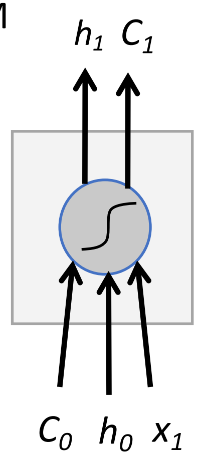
$$e_t = -\log(y_t(GT_t))$$

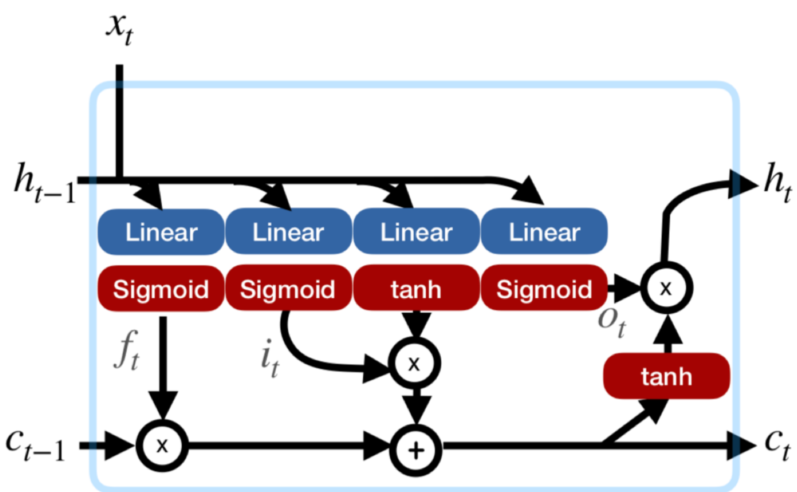
$$y_t = \text{softmax}(W_y h_t)$$

$$h_t = \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

------ shared weights

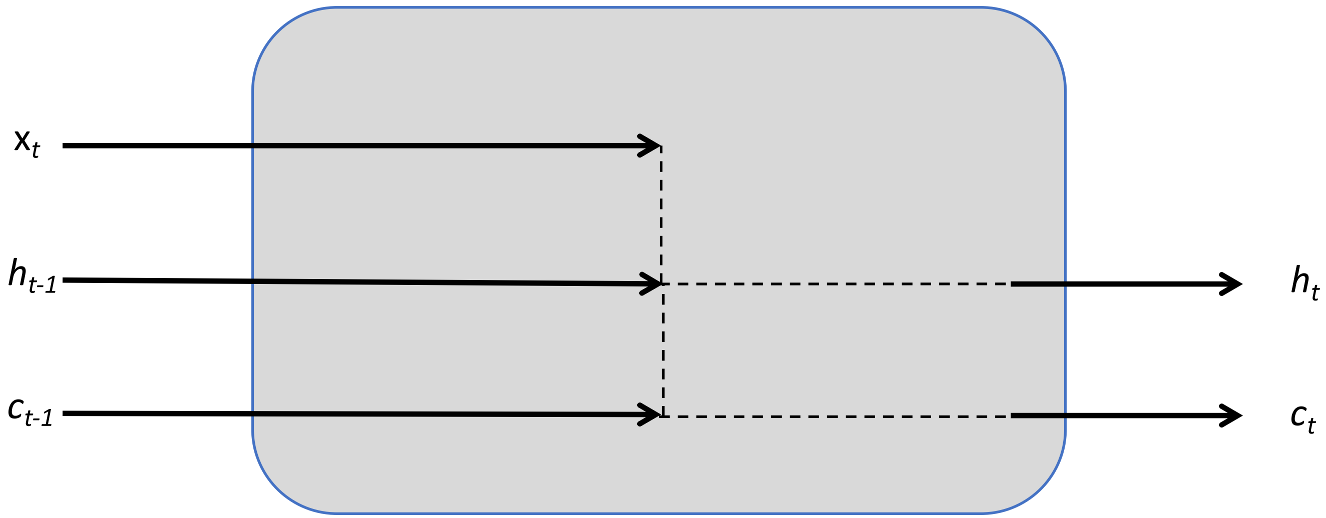
LSTM





Long Short-Term Memory (LSTM)

- Add a *memory cell* that is not subject to matrix multiplication or squishing, thereby avoiding gradient decay



S. Hochreiter and J. Schmidhuber, Long short-term memory, Neural Computation 9 (8), pp. 1735–1780, 1997

- RNN harder to train – more specifics
- For many of the tasks temporal models are
- Pros of LSTMs
- - variable input sequenecs
- - variable output sequences
- Hard to train – problems with modelling longer sequences (beyond ~ 100 steps)

  - both the gate, memory cell models

An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, Bai etal., arXiv 2018

Temporal models from processing video

- Relevant for:
- Action recognition
- Tracking
- Monocular motion estimation
- Future prediction

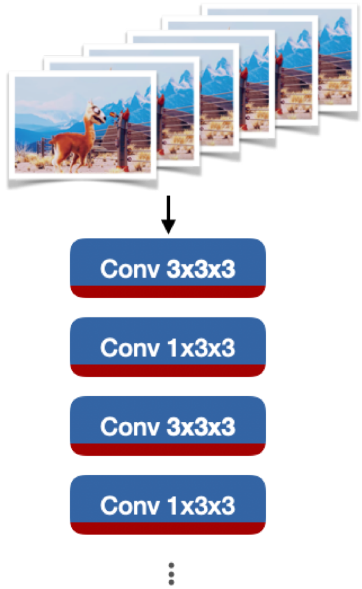
1. Keep frames independent
2. Temporal model fit RNN

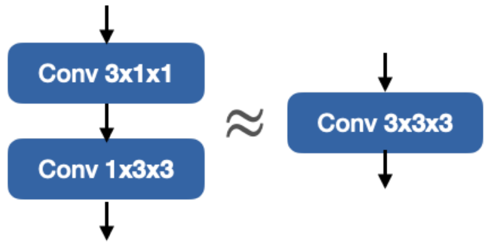
Temporal convolutions

- 3D CNN
- Add temporal dimension to convolution
- 3D kernel
- Slide the kernel through space and time
- $w \times h \times 3 \times t$
- Too many parameters, too slow
- Idea separate spatial and temporal convolutions (C is number of channels)

$O((WH+T) \times C^2)$

$O(WHT) \times C^2)$



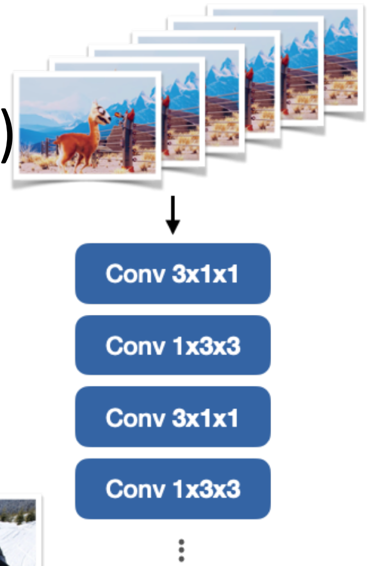


Temporal models – Activity Recognition

- Train represention on Image Net 2D CNN
- Inflate the network in time (replicated the kernel in time)
- Or insert 1D temporal convolutions initialized by

(averaging and fine tune on video dataset), faster training

still computationally demanding



Quo Vadis Action Recognition, Carreira and Zisserman, 2017

Computer Vision Problems

- Activities of daily living
- Relationship between space and time

- Representations obtained by vision algorithms

often do not, results imperfect and often not all relevant

For the task at hand, e.g. Autonomous Driving

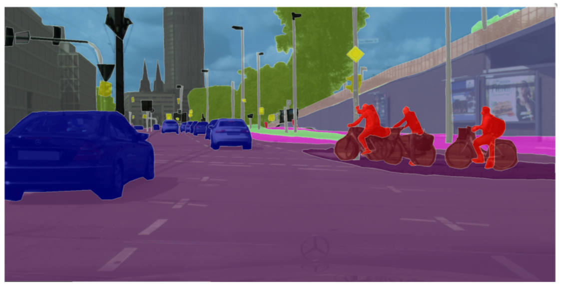
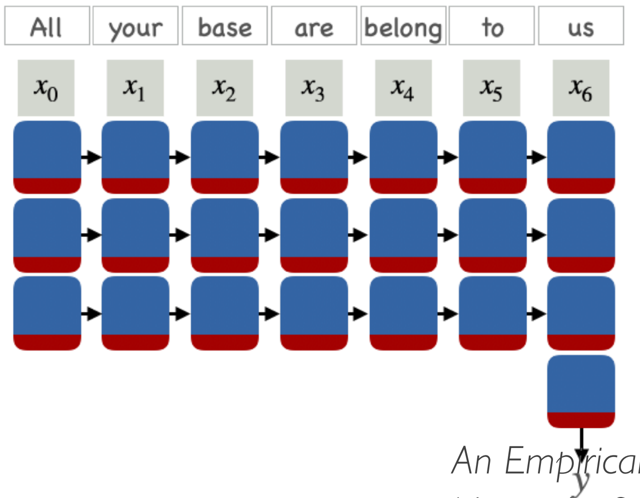
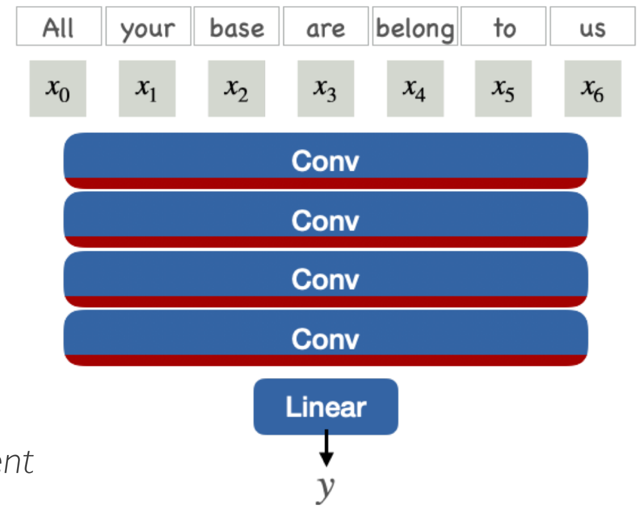


Image source: EPIC-Kitchens dataset, Damen et al., https://arxiv.org/abs/1804.02748

Temporal models in general

- Recurrent models – input sequence output – predicts output – for next word prediction task
- We can handle the variable length – use convolutions (dilated convolutions – the receptive field grows with the number of layers)
- How to handle temporal models for the same task





An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, Bai etal., arXiv 2018

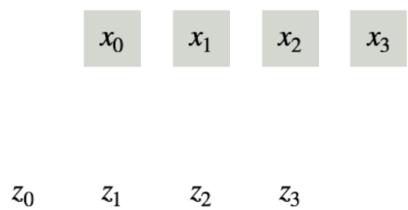
Temporal convolutions

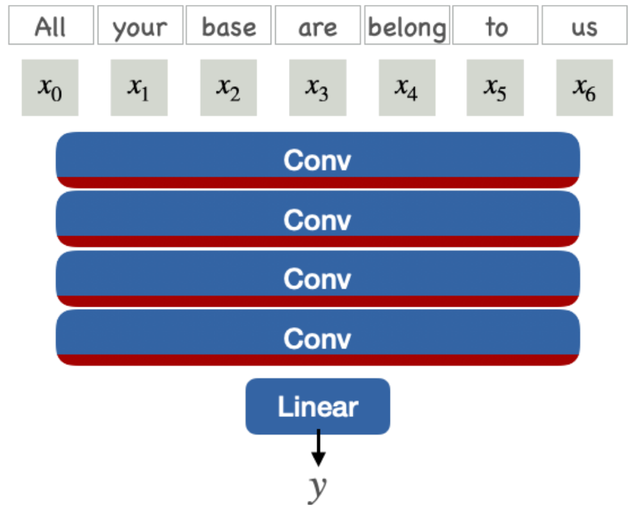
- Dilated convolutions
- 5-10 layers – 100 steps context

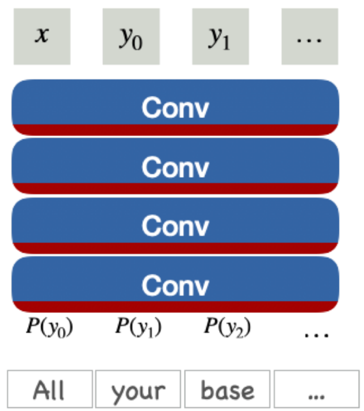
- Caveat - need causal convolutions
- Only look in the past (works in 1D)
- Autoregressive model

$$P(y_0 \mid x) \cdot P(y_1 \mid x, y_0) \cdot P(y_2 \mid x, y_0, y_1) \cdot \ldots$$

- Shift input

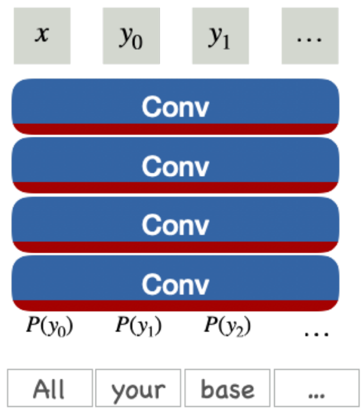


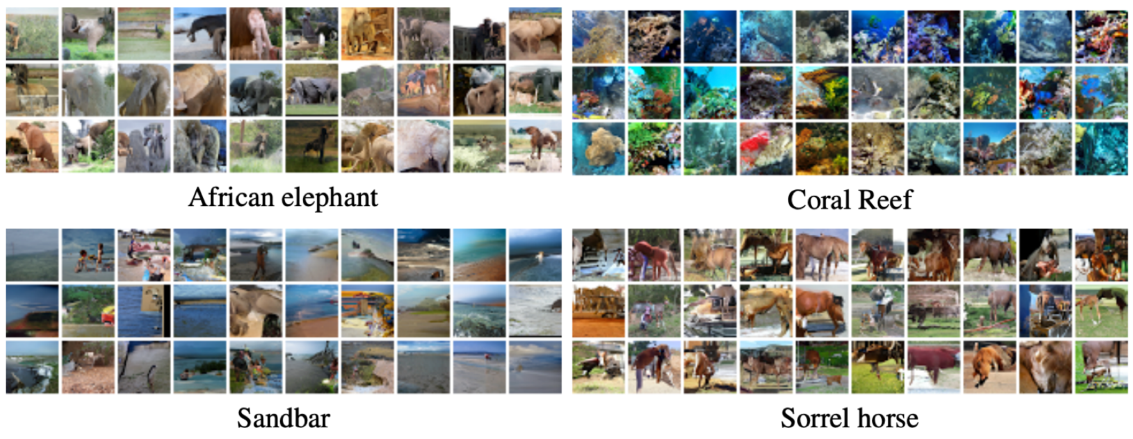




- How to generate sequences ?
- Need to modify convolutions to look in the past not the future, harder for images

New type of decoder conditioned on image encoder

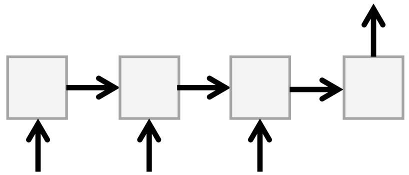




Conditional image generation with pixelCNN decoders, [Aaron van den Oord](), [Nal Kalchbrenner](), [Oriol Vinyals](), [Lasse Espeholt](), [Alex Graves](), [Koray Kavukcuoglu]()

RNN use Cases

Multiple input –
Single output



Sequence Classification

Single - Multiple

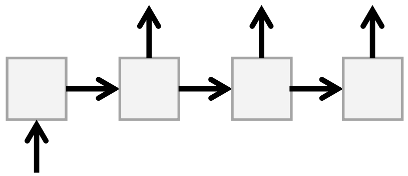


Image Captioning

Multiple - Multiple

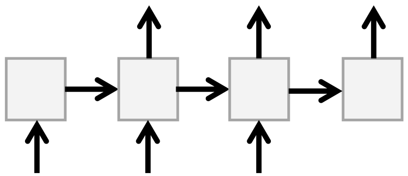
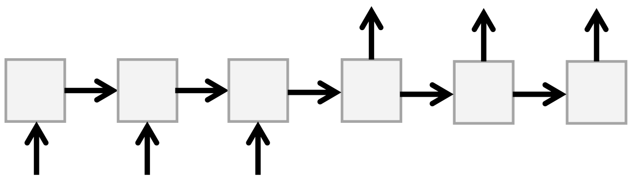


Image Captioning

Multiple - Multiple



Translation

Review: Image captioning

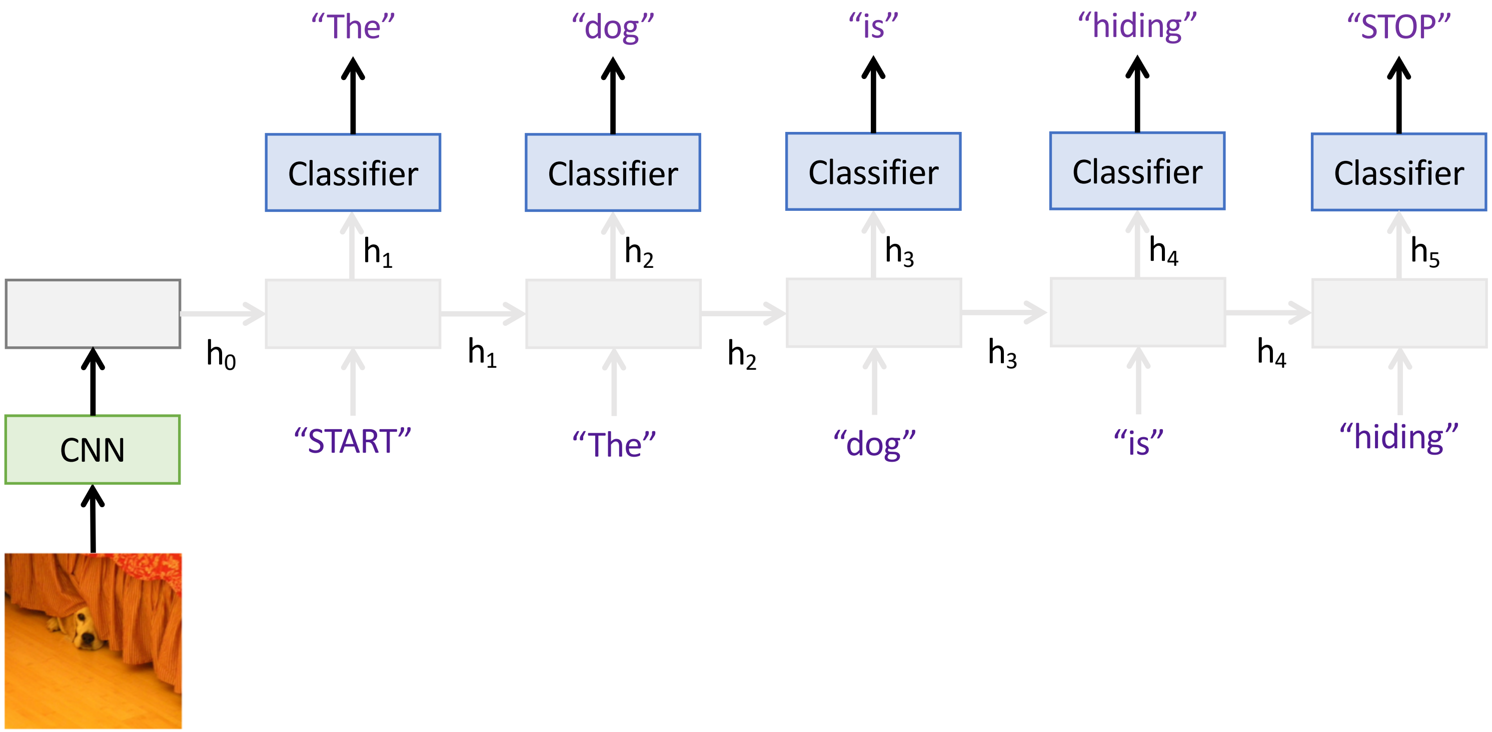
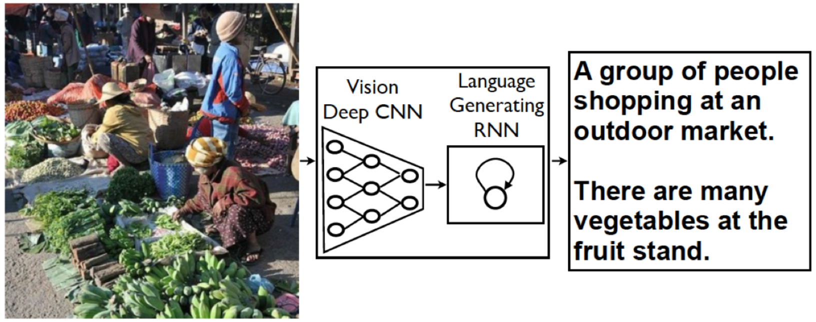
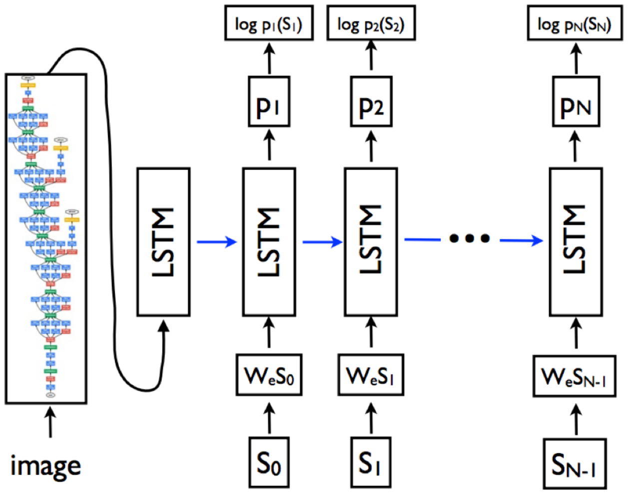


Image Caption Generation





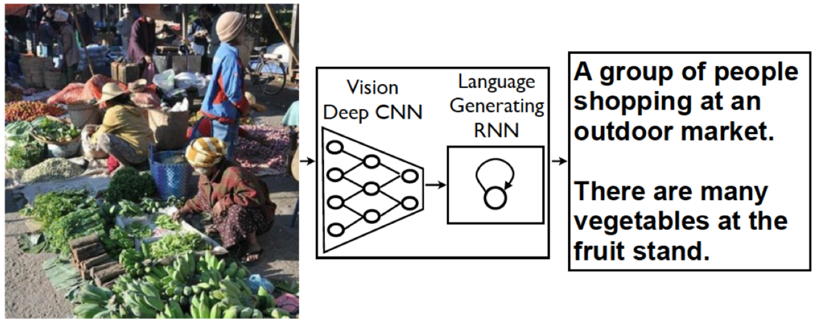
$$
\begin{aligned}
x_{-1} &= \mathrm{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0 \ldots N-1\} \\
p_{t+1} &= \mathrm{LSTM}(x_t), \quad t \in \{0 \ldots N-1\}
\end{aligned}
$$

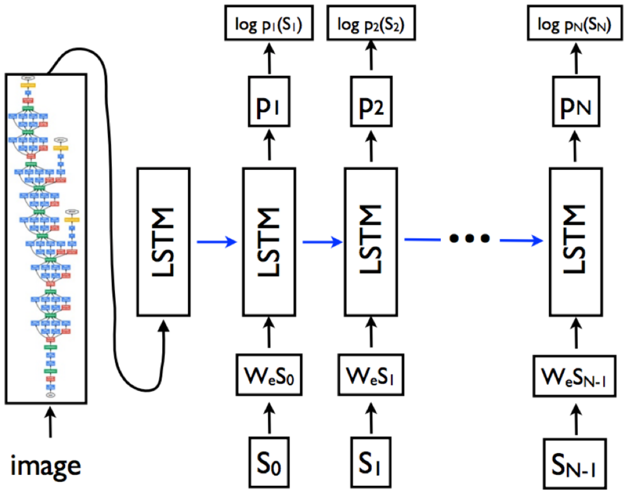
$$
\log p(S|I) = \sum_{t=0}^{N} \log p(S_t | I, S_0, \ldots, S_{t-1})
$$

$$
L(I, S) = -\sum_{t=1} \log p_t(S_t)
$$

O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and Tell: A Neural Image Caption Generator, CVPR 2015

Image Caption Generation





$$
\begin{aligned}
x_{-1} &= \text{CNN}(I) \\
x_t &= W_e S_t, \quad t \in \{0 \dots N-1\} \\
p_{t+1} &= \text{LSTM}(x_t), \quad t \in \{0 \dots N-1\}
\end{aligned}
$$

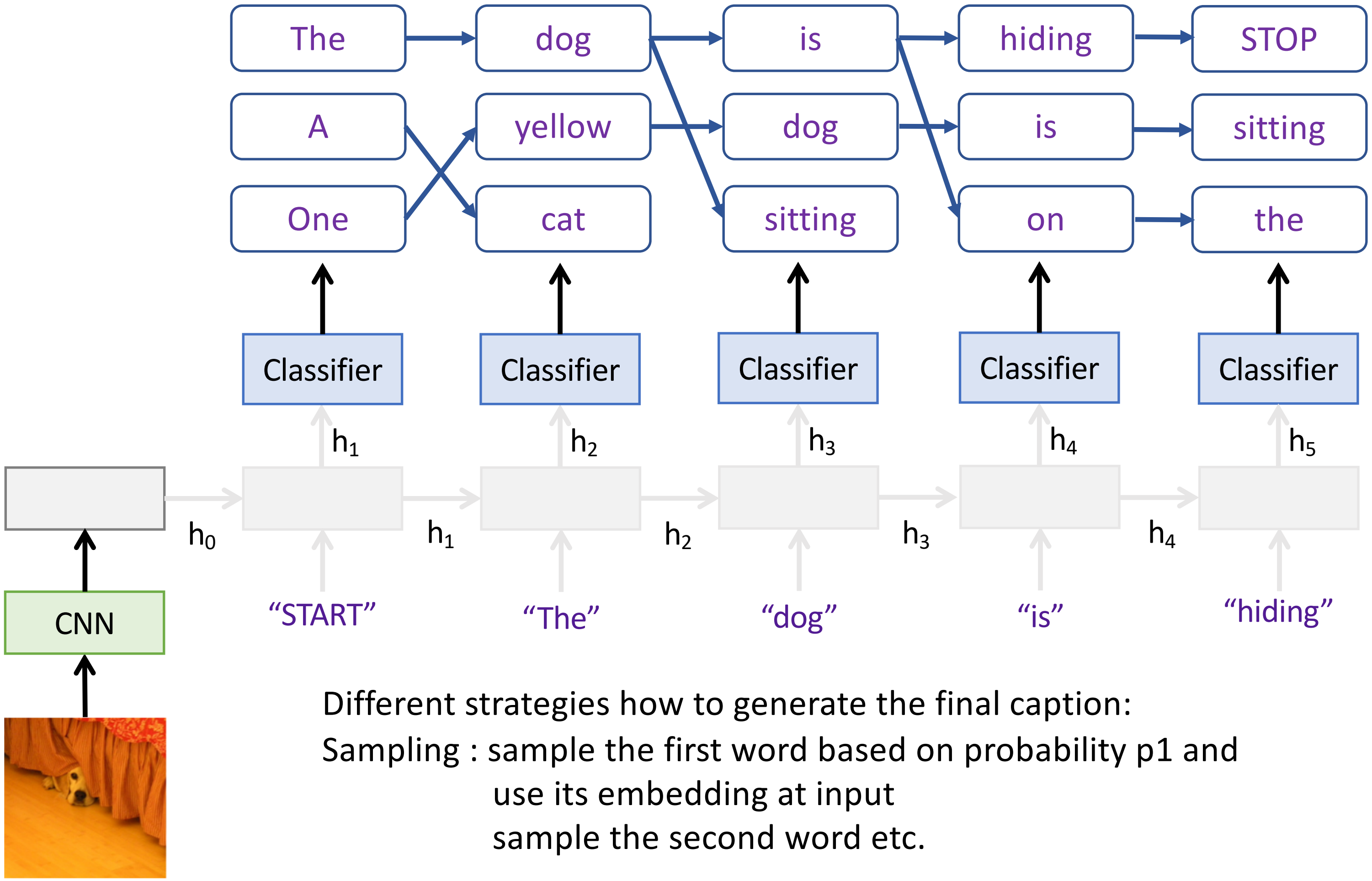
$$\log p(S|I) = \sum_{t=0}^{N} \log p(S_t | I, S_0, \dots, S_{t-1})$$

$$L(I, S) = -\sum_{t=1} \log p_t(S_t)$$ training

O. Vinyals, A. Toshev, S. Bengio, D. Erhan, Show and Tell: A Neural Image Caption Generator, CVPR 2015

Image Caption Generation

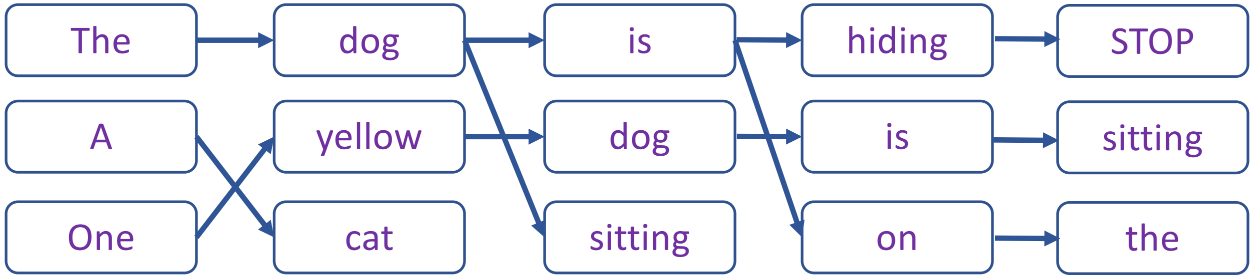
Review: Image captioning



Different strategies how to generate the final caption:

Sampling : sample the first word based on probability p1 and
use its embedding at input
sample the second word etc.

Beam search



- Maintain *k* top-scoring candidate sentences (according to sum of per-word log-likelihoods)
  - At each step, generate all their successors and reduce to *k* (*beam width*)

How to evaluate image captioning?

- Reference sentences (written by human annotators):
  - “A dog hides underneath a bed with its face peeking out of the bed skirt”
  - “The small white dog is peeking out from under the bed”
  - “A dog is peeking its head out from underneath a bed skirt”
  - “A dog peeking out from under a bed”
  - “A dog that is under a bed on the floor”

- Generated sentence:
  - “A dog is hiding”

BLEU: Bilingual Evaluation Understudy

- **N-gram precision**: count the number of n-gram matches between candidate and reference translation, divide by total number of n-grams in candidate translation
  - Clip counts by the maximum number of times an n-gram occurs in any reference translation
  - Multiply by *brevity penalty* to penalize short translations

- Most commonly used measure despite well-known shortcomings
- *H(i)* number of i-gram tuples, *Matched(i)* is number of times tuple occurs in hypothesis, min with number of times tuple occurs in reference

$$P(i) = \frac{Matched(i)}{H(i)} \qquad Matched(i) = \sum_{t_i} \min\{C_h(t_i), \max_j C_{hj}(t_i)\}$$

$$BLEU_a = \{\prod_{i=1}^{N} P(i)\}^{1/N}$$

K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, ACL 2002

Microsoft
Common Objects in Context

cocodataset@outlook.com

Home People Explore **Dataset**

Overview | Challenges | Download | Evaluate | Leaderboard

Table-C5 | Table-C40 | 2015 Captioning Challenge

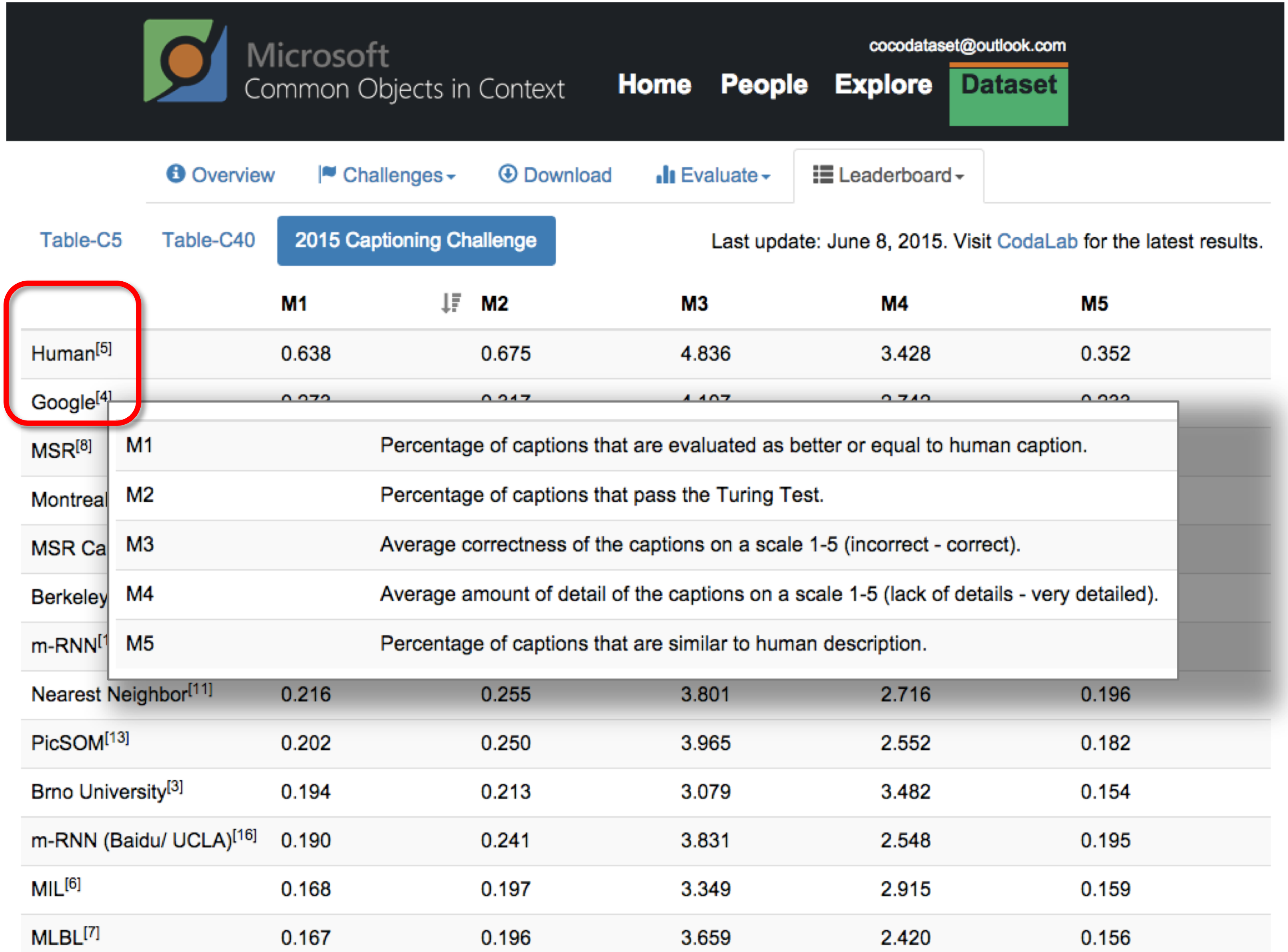
Last update: June 8, 2015. Visit CodaLab for the latest results.

	CIDEr-D	Meteor	ROUGE-L	BLEU-1	BLEU-2	BLEU-3	BLEU-4
m-RNN (Baidu/ UCLA)[16]	0.886	0.238	0.524	0.72	0.553	0.41	0.302
m-RNN[15]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.299
MSR Captiva	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.308
Google[4]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.309
Berkeley LR	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.277
Nearest Neig	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.28
MSR[8]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	.291
Montreal/Toronto[10]	0.85	0.243	0.513	0.689	0.515	0.372	0.268
PicSOM[13]	0.833	0.231	0.505	0.683	0.51	0.377	0.281
Tsinghua Bigeye[14]	0.673	0.207	0.49	0.671	0.494	0.35	0.241
MLBL[7]	0.74	0.219	0.499	0.666	0.498	0.362	0.26
Human[5]	0.854	0.252	0.484	0.663	0.469	0.321	0.217

Metrics

CIDEr-D	CIDEr: Consensus-based Image Description Evaluation
METEOR	Meteor Universal: Language Specific Translation Evaluation for Any Target Language
Rouge-L	ROUGE: A Package for Automatic Evaluation of Summaries
BLEU	BLEU: a Method for Automatic Evaluation of Machine Translation

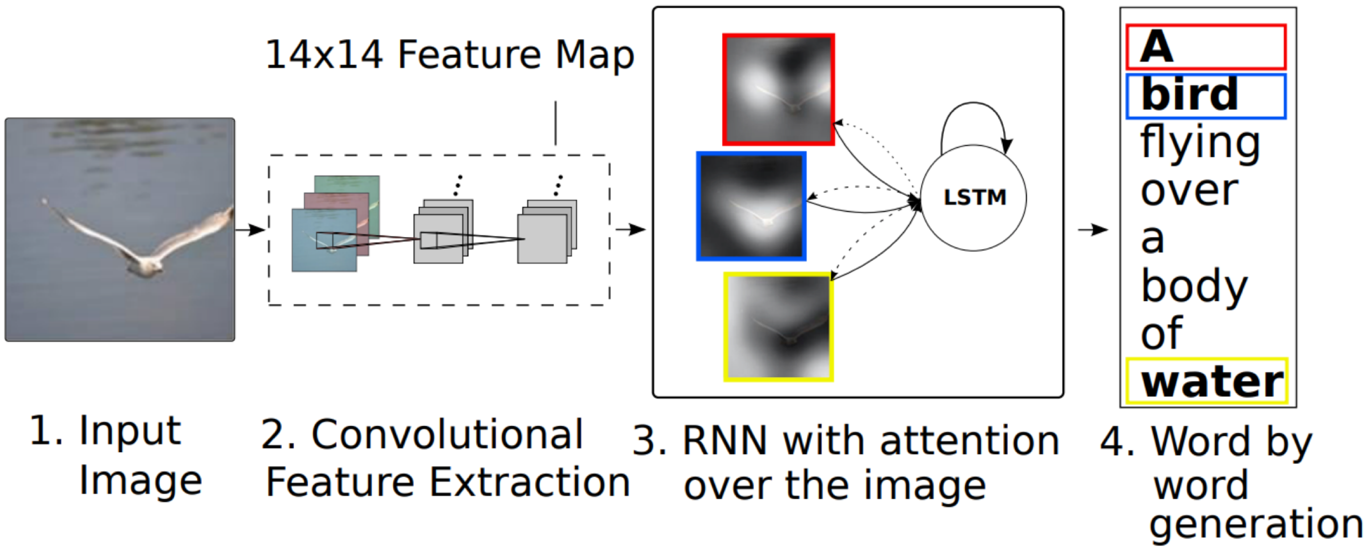
http://mscoco.org/dataset/#captions-leaderboard



	M1	M2	M3	M4	M5
Human[5]	0.638	0.675	4.836	3.428	0.352
Google[4]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MSR[8]					
Montreal					
MSR Ca					
Berkeley					
m-RNN[1					
Nearest Neighbor[11]	0.216	0.255	3.801	2.716	0.196
PicSOM[13]	0.202	0.250	3.965	2.552	0.182
Brno University[3]	0.194	0.213	3.079	3.482	0.154
m-RNN (Baidu/ UCLA)[16]	0.190	0.241	3.831	2.548	0.195
MIL[6]	0.168	0.197	3.349	2.915	0.159
MLBL[7]	0.167	0.196	3.659	2.420	0.156

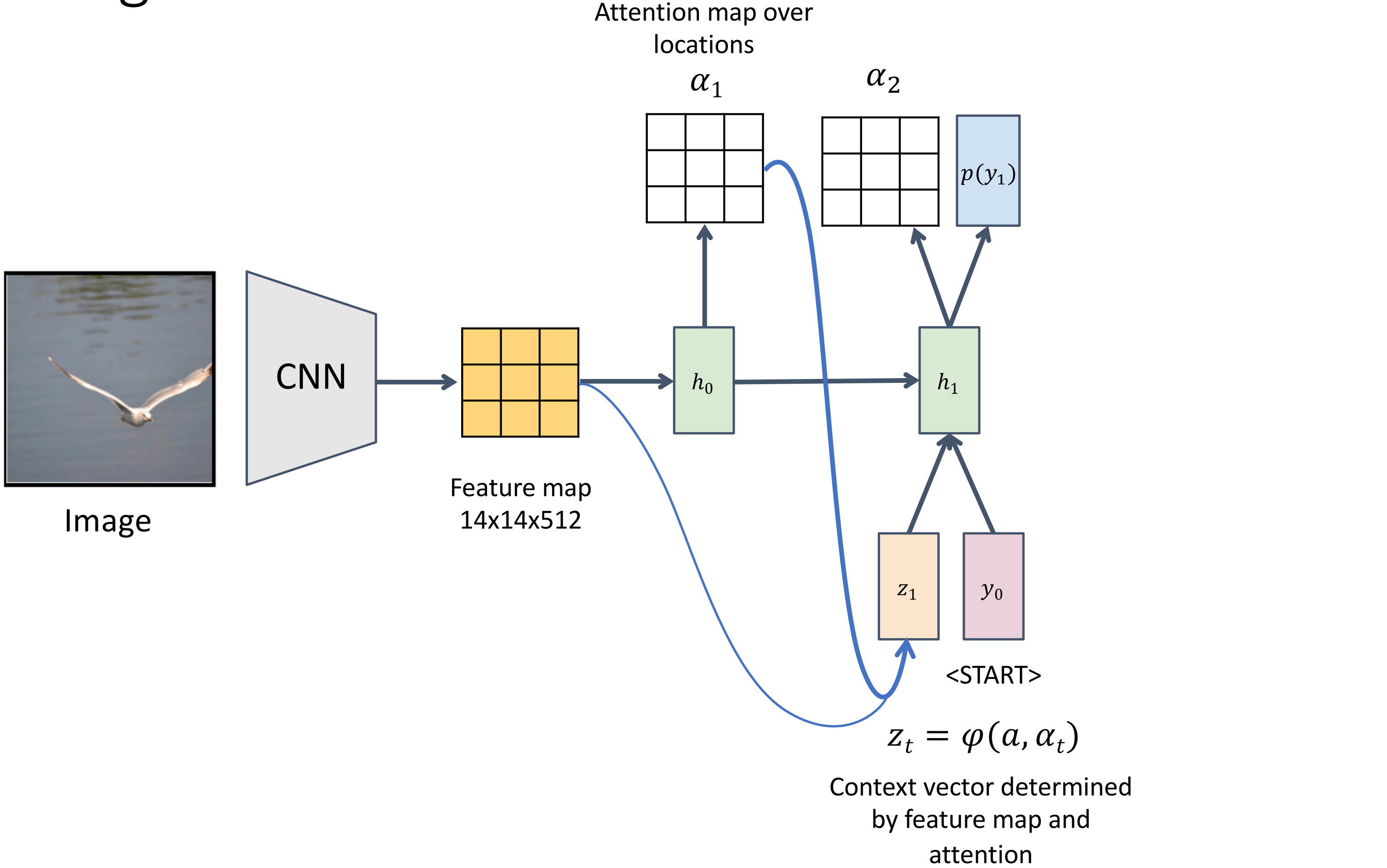
M1	Percentage of captions that are evaluated as better or equal to human caption.
M2	Percentage of captions that pass the Turing Test.
M3	Average correctness of the captions on a scale 1-5 (incorrect - correct).
M4	Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed).
M5	Percentage of captions that are similar to human description.

Captioning with attention



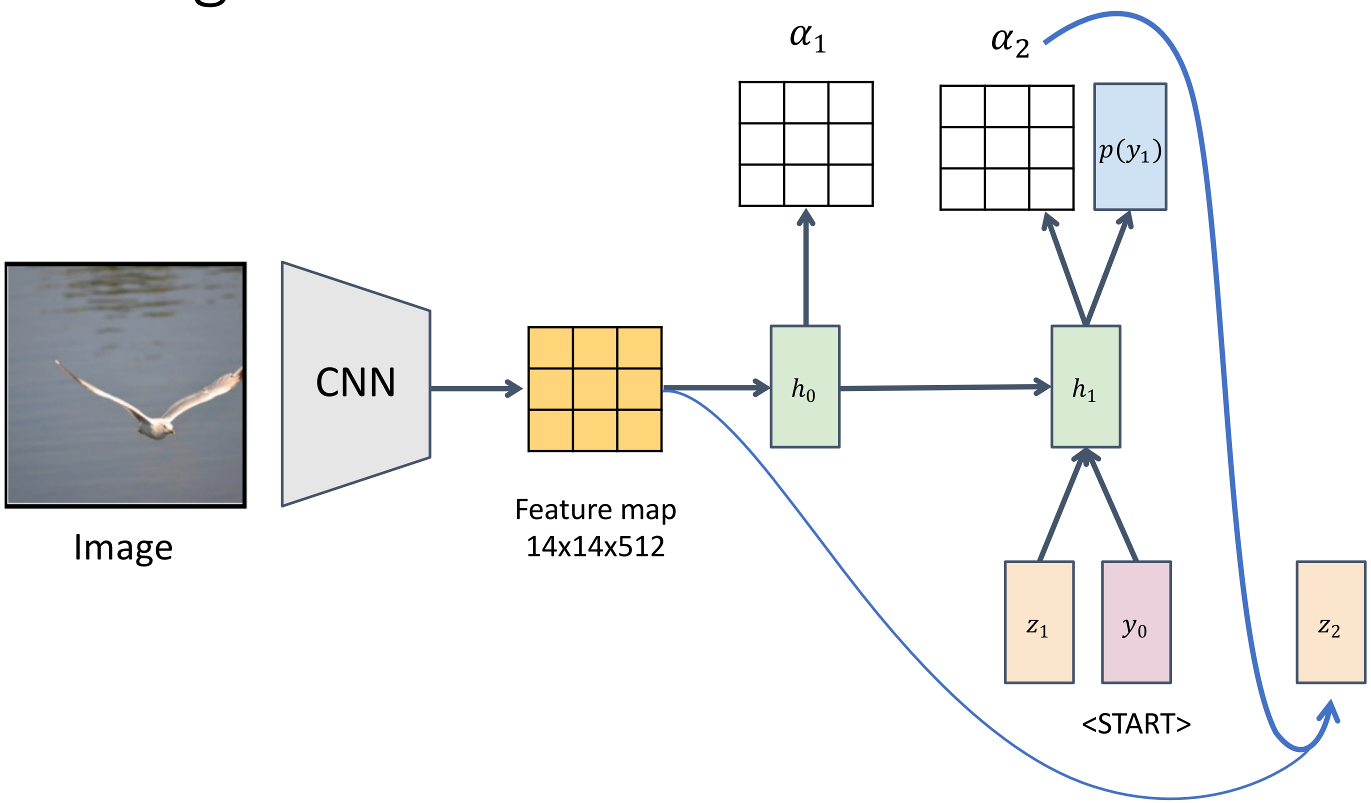
K. Xu et al., [Show, Attend and Tell: Neural Image Caption Generation with Visual Attention](), ICML 2015

Captioning with attention



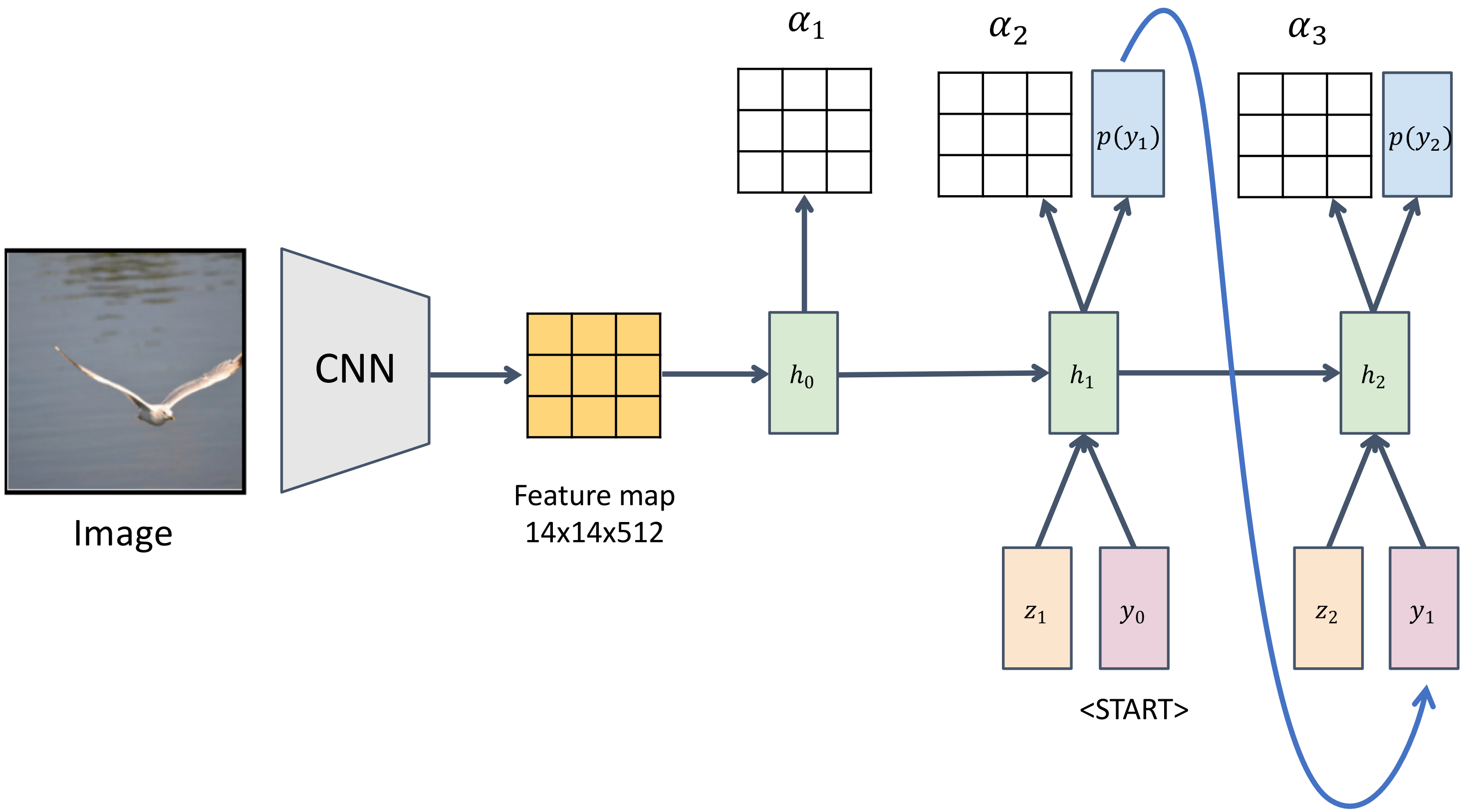
Adapted from Stanford CS231n, Berkeley CS294

Captioning with attention



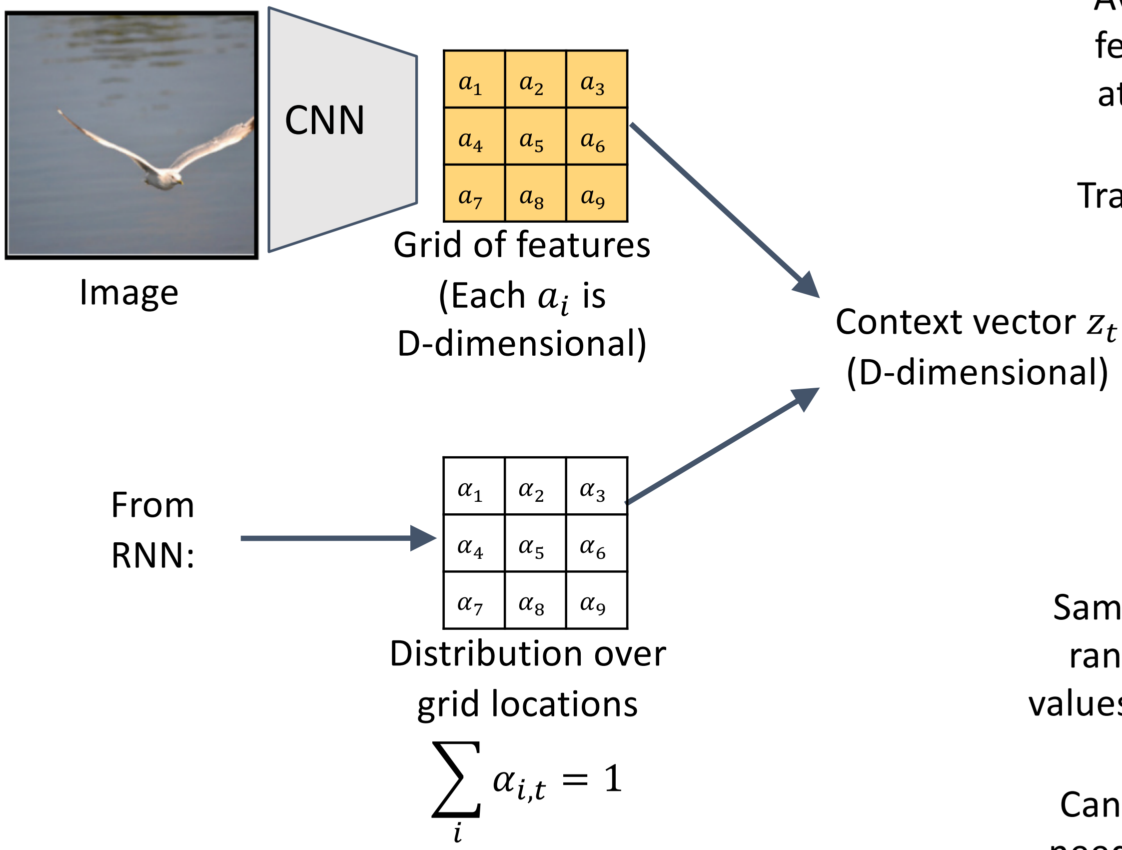
Adapted from [Stanford CS231n](), [Berkeley CS294]()

Captioning with attention



Adapted from Stanford CS231n, Berkeley CS294

“Soft” and “hard” attention



Soft attention:
Average over locations of feature map weighted by attention: $z_t = \sum_i \alpha_{i,t} \, a_i$

Train with gradient descent

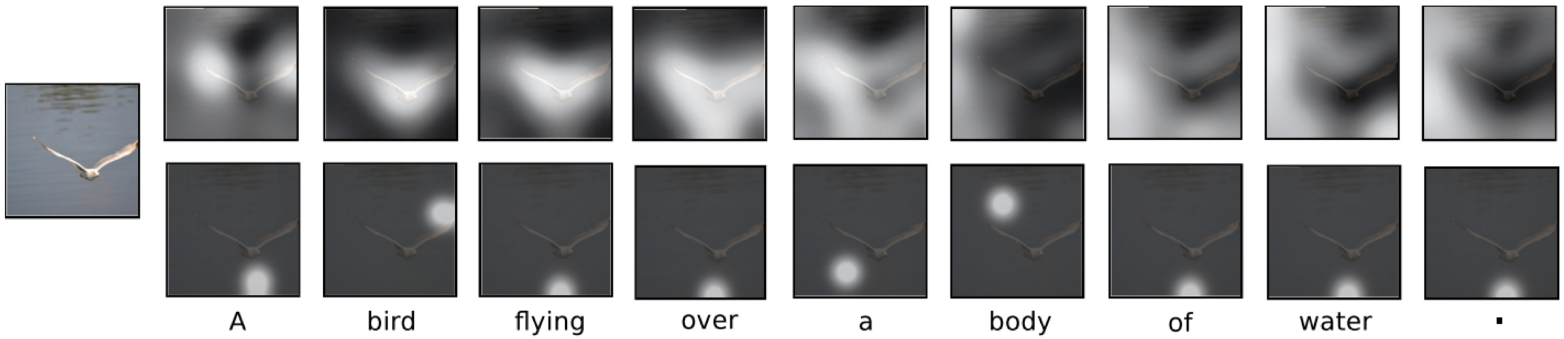
Hard attention:
Sample ONE location: z_t is a random variable taking on values a_i with probabilities $\alpha_{i,t}$

Can’t use gradient descent; need reinforcement learning

Adapted from Berkeley CS294

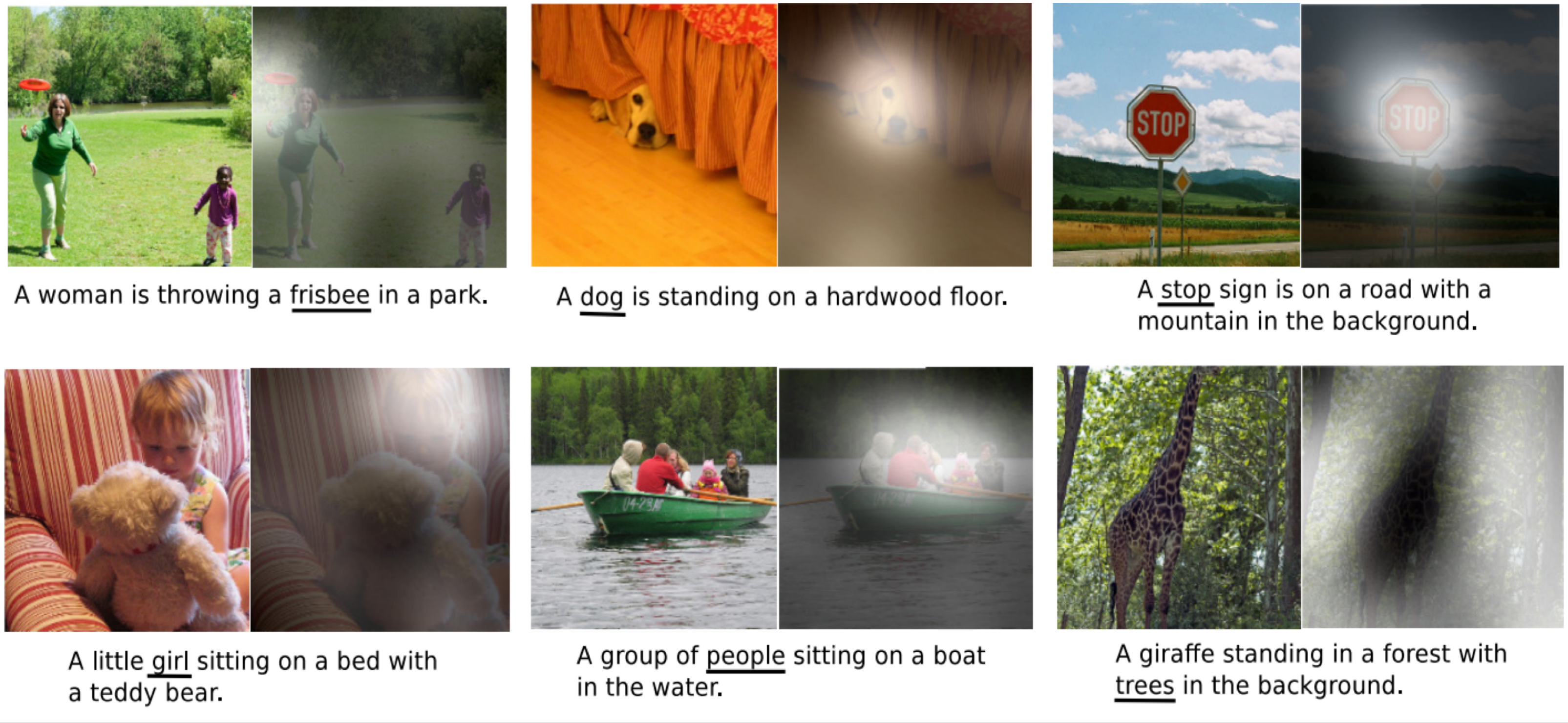
"Soft" and "hard" attention

Soft attention:
Average over locations of feature map weighted by attention: $z_t = \sum_i \alpha_{i,t}\, a_i$



Hard attention:
Sample ONE location: z_t is a random variable taking on values a_i with probabilities $\alpha_{i,t}$

Example Results



Example Results

- M

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

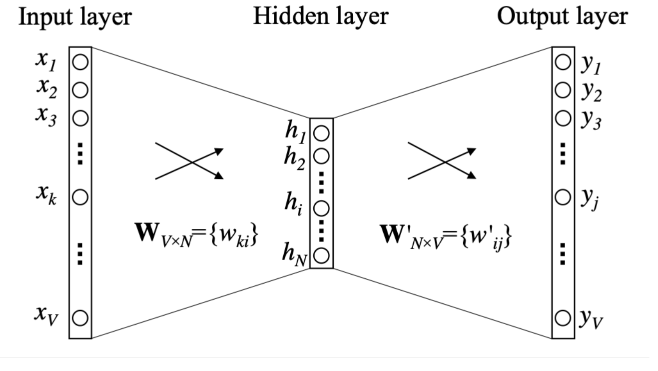
A man is talking on his cell phone
while another man watches.

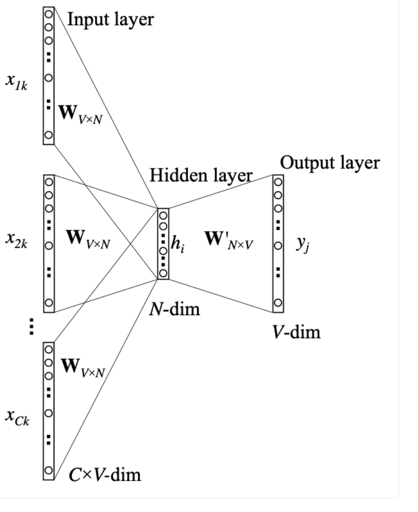
Word embeddings

- Mikolov – how to learn word representations
- Skip gram model – learn quality representations of words from unstructured text
- vec("Madrid") - vec("Spain") + vec("France") is closer to vec("Paris")
- Distributed Representations of Words and Phrases and their Compositionality
- Idea – how to capture the similarity between words instead of treating them as atomic units
- Train a network to generate vector represntations of words

word2vec

- Word2vec parameter learning explained Xin Rong https://arxiv.org/abs/1411.2738
- bit.ly/wevi-online
- Variations – one word to one – hidden layer is the embedding
- N-gram model to one
- Other alternative GloVec





RNN vs Sequence to Sequence models

- Especially when it comes to seq2seq models, is one hidden state really enough to capture global information pertaining to the translation?
- Idea – learn a context vector – for each input vector we learn a set of weights of how much the remained the sentence is affected by the rest of the sentence
- Sequence attention model – more detail next