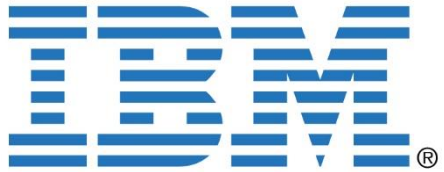
Multilingual Multimodal Language Processing Using Neural Networks

Mitesh M Khapra
IBM Research India

Sarath Chandar
Université de Montréal



Multilingual Multimodal Language Processing Using Neural Networks

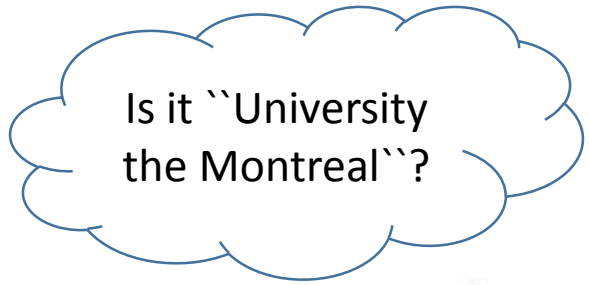
Mitesh M Khapra
IBM Research India

Sarath Chandar
Université de Montréal

NAACL HLT 2016 San Diego

IBM

Université de Montréal



Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
5. Summary and open problems

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
5. Summary and open problems

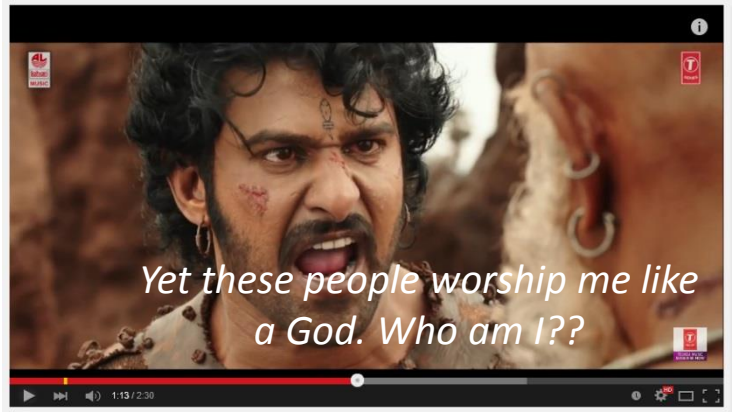
What is multilingual multimodal NLP?

Designing language processing systems that can handle

- Multiple languages (English, French, German, Hindi, Spanish,…)
- Multiple modalities (image, speech, video,…)

Why multilingual multimodal NLP?

We live an increasingly multilingual multimodal world



Video, Tamil Audio, English subtitles

English: brown horses eating tall grass beside a body of water

French: chevaux brun manger l'herbe haute à côté d'un corps de l'eau

Why multilingual multimodal NLP?

[en] Approval for *pup* **rescue**
A tired *seal pup* which refused to leave an oil industry ship 100 miles out to sea is recovering before being released back into the wild.

[pt]Aprovação para resgate *filhote de cachorro*
Um *selo* cansado, que se recusou a sair de um navio indústria petrolífera 100 milhas para o mar está se recuperando antes de ser liberado de volta na natureza.

Seal – selo (stamp) and foca (marine animal).

Seal pup should have been translated to filhote de foca (young seal), but it has been translated as selo.

Pup in title has been wrongly translated to filhote de cachorro (young dog).

*This example is taken from Calixto et al., 2012.

Why multilingual multimodal NLP?

How Facebook Is Helping Blind People 'See' Photos

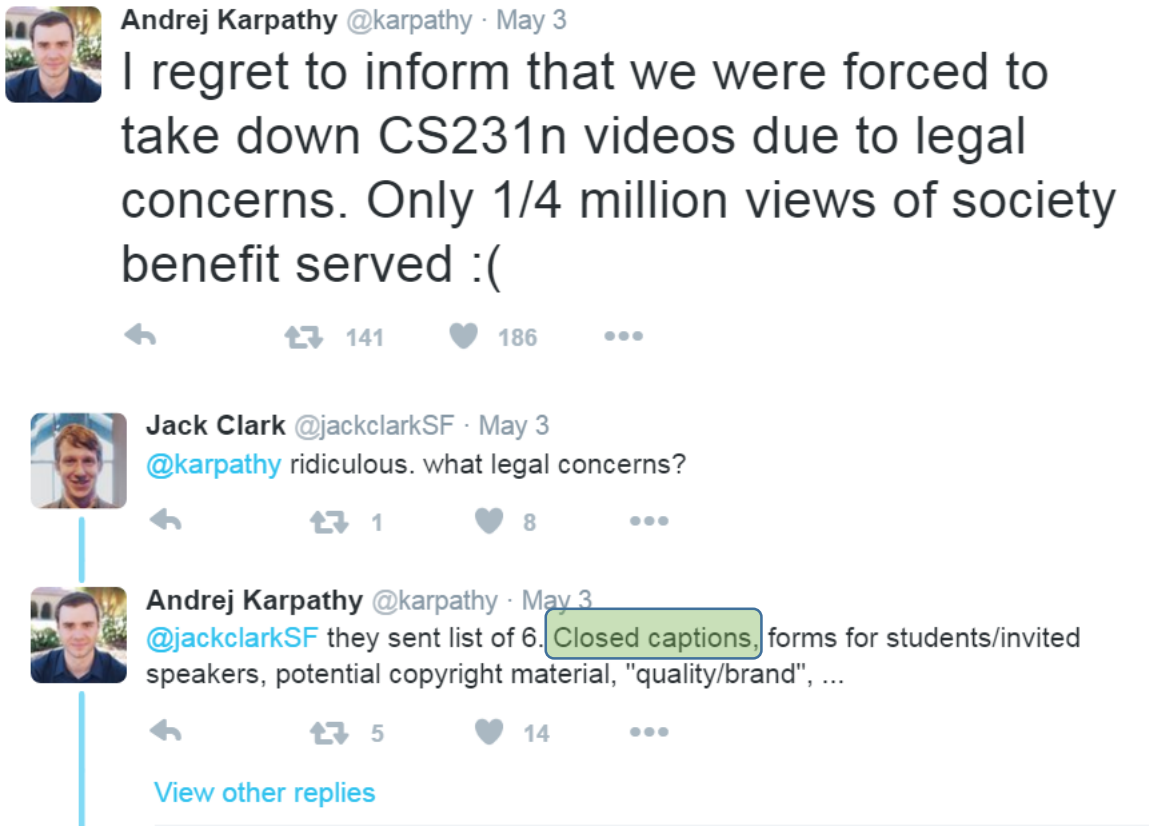
Amit Chowdhry, CONTRIBUTOR

I cover noteworthy technology, startups and gadgets **FULL BIO**

Opinions expressed by Forbes Contributors are their own.

*Article from forbes.com

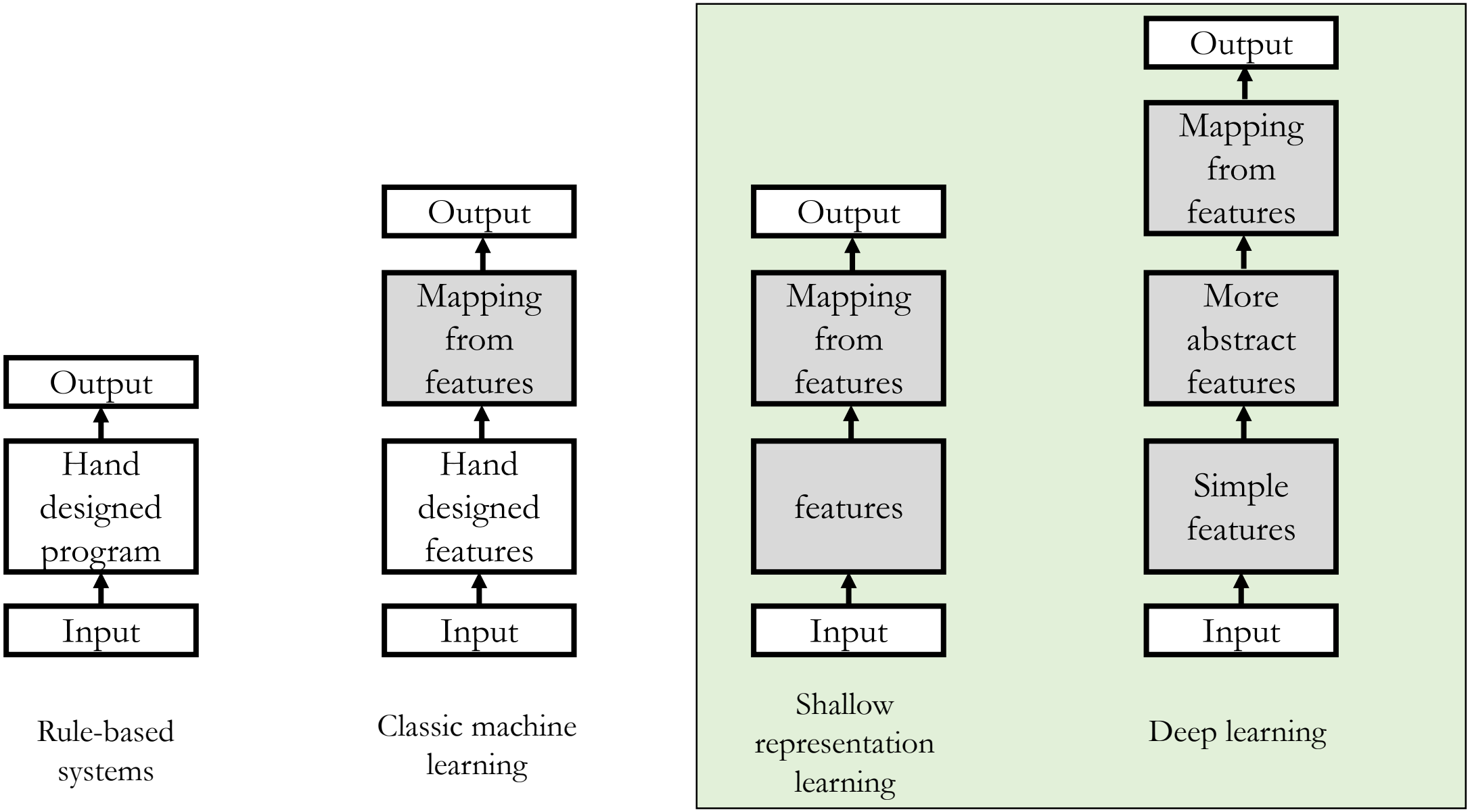
Why multilingual multimodal NLP?



Why using Neural Networks?

- *Backpropaganda* of Neural Networks in the name of Deep Learning.
- Significant success in speech recognition, computer vision.
- Slowly conquering NLP?

Deep Representation Learning



Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
   a) Neural network and backpropagation
   b) Matching data with architectures
   c) Auto-encoders
   d) Distributed natural language processing
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
5. Summary and open problems

Artificial Neuron / Perceptron

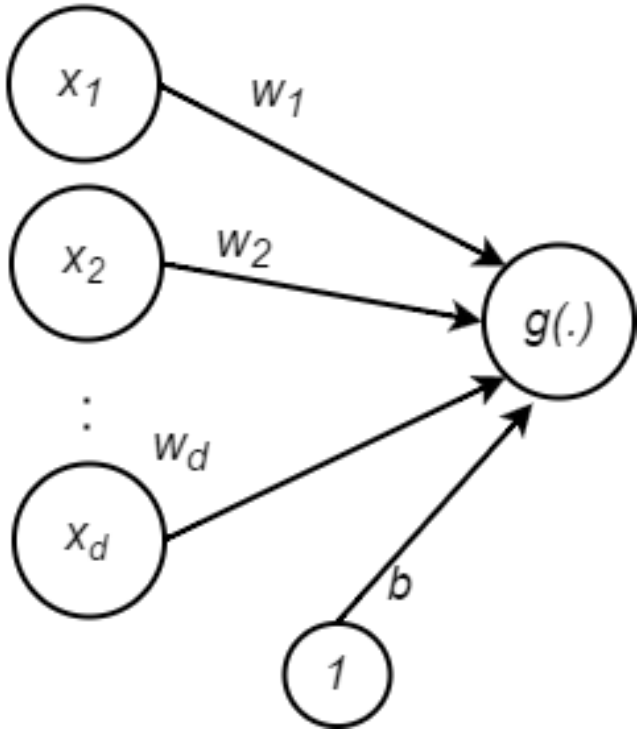
Neuron pre-activation:

$$a(x) = b + \sum_i w_i x_i = b + w^T x$$

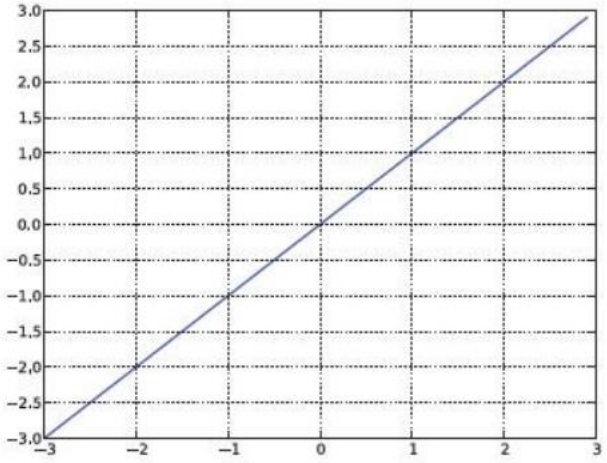
Neuron activation:

$$o = g(a(x)) = g(b + w^T x)$$

- W – connection weights
- b – neuron bias
- $g(.)$ – activation function

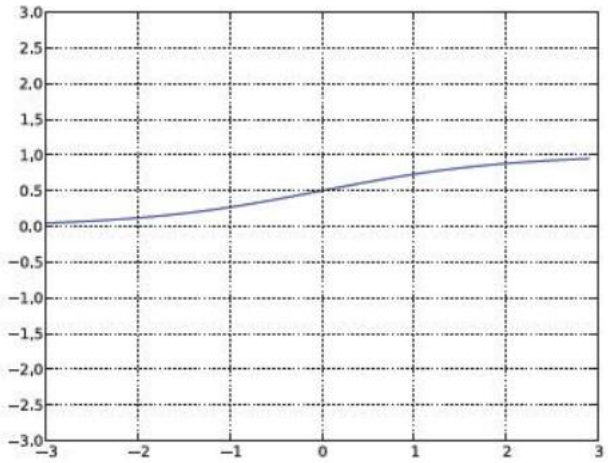


Activation functions



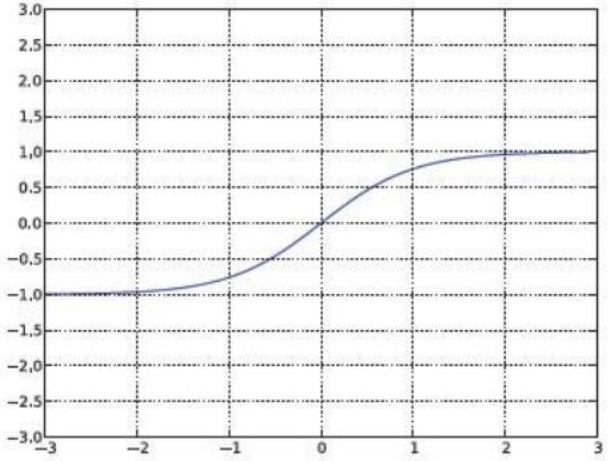
Identity

$$g(a) = a$$



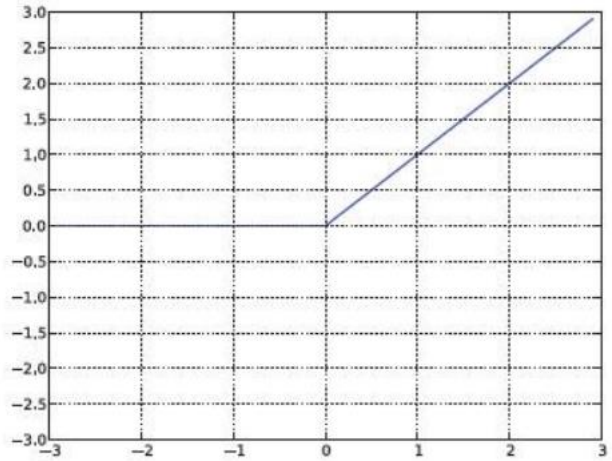
sigmoid

$$g(a) = sigm(a) = \frac{1}{1 + exp(a)}$$



tanh

$$g(a) = tanh(a) = \frac{exp(2a) - 1}{exp(2a) + 1}$$



relu

$$g(a) = relu(a) = max(0, a)$$

*Images from Hugo Larochelle's course.

Learning problem

Given training data $\mathcal{D} = \{x^{(i)}, y^{(i)}\}_{i=1}^{N}$ find W and b that minimizes

$$\mathcal{J}(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (g(a(x^{(i)})) - y^{(i)})^2$$

$\theta = \{W, b\}$ are the parameters of the perceptron model.

Learning using gradient descent

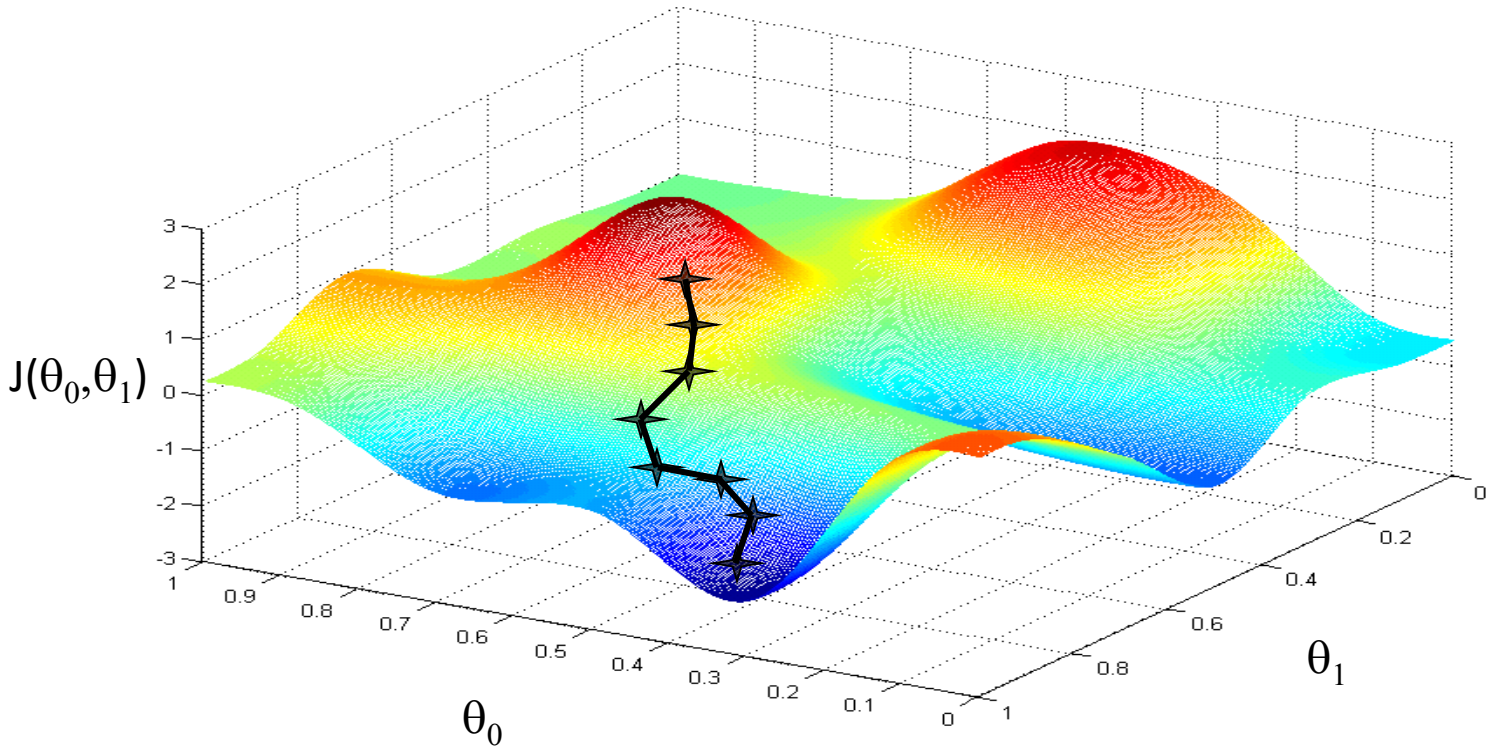
- Compute gradient w.r.t the parameters.
- Make a small step in the direction of the negative gradient in the parameter space.

$$W = W - \alpha \frac{\partial \mathcal{J}}{\partial W}$$

$$b = b - \alpha \frac{\partial \mathcal{J}}{\partial b}$$

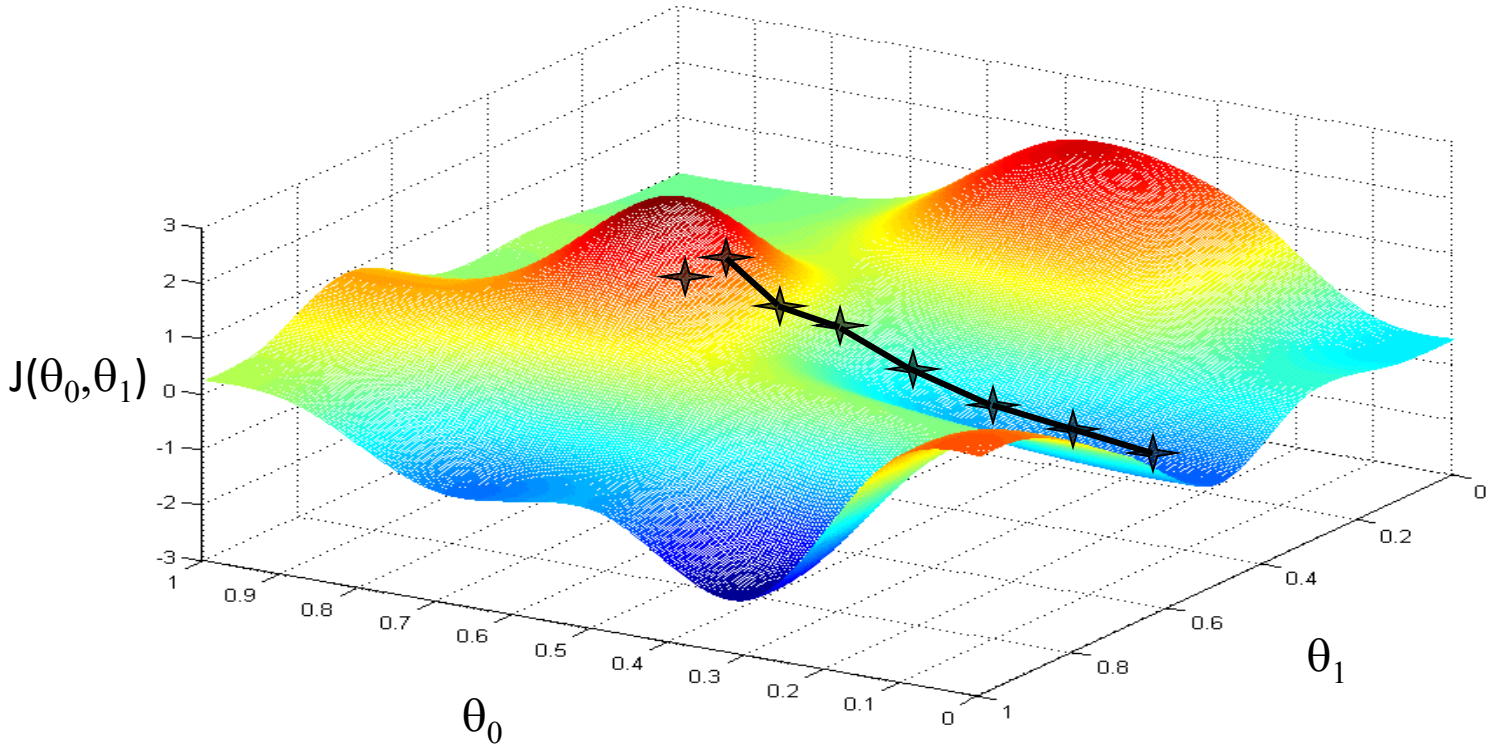
α - learning rate or step size.

Gradient Descent



*This animation is taken from Andrew Ng's course.

Gradient Descent



*This animation is taken from Andrew Ng's course.

Stochastic Gradient Descent (SGD)

- Approximate the gradient using a mini-batch of examples instead of entire training set.
- Online SGD – when mini-batch size is 1.
- SGD is most commonly used when compared to full-batch GD.

Online SGD for Perceptron Learning

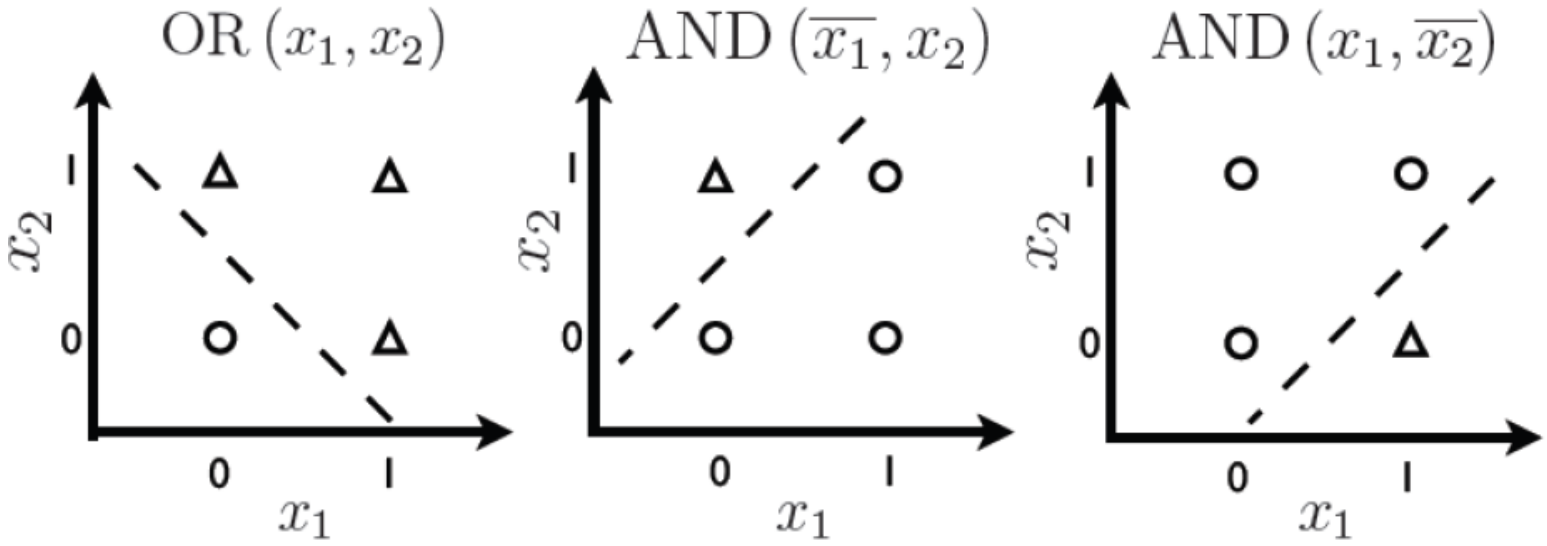
$$w_k = w_k - \alpha \; . \; (g(a(x^{(i)}) - y^{(i)}) \; . \; g'(a(x^{(i)})) \; . \; x_k^{(i)} \qquad for \; k = 1, .., d$$

$$b = b - \alpha \; . \; (g(a(x^{(i)}) - y^{(i)}) \; . \; g'(a(x^{(i)}))$$

- Perceptron learning objective is a convex function.
- GD is guaranteed to converge to global minimum while SGD will converge to global minimum by slowly letting learning rate decrease to zero.

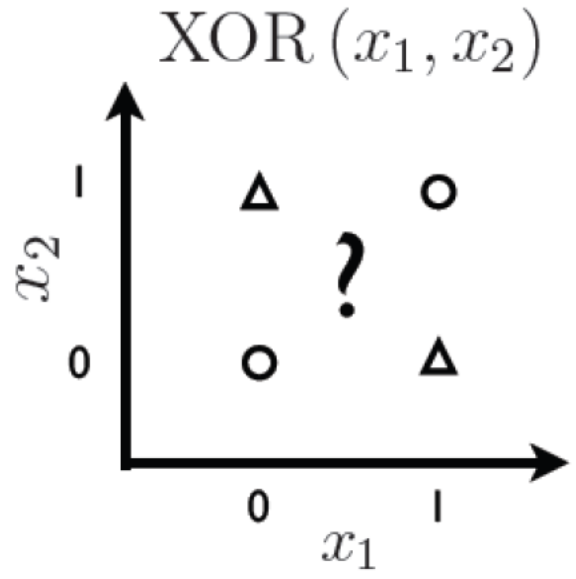
What can a perceptron do?

- Can solve linearly separable problems.



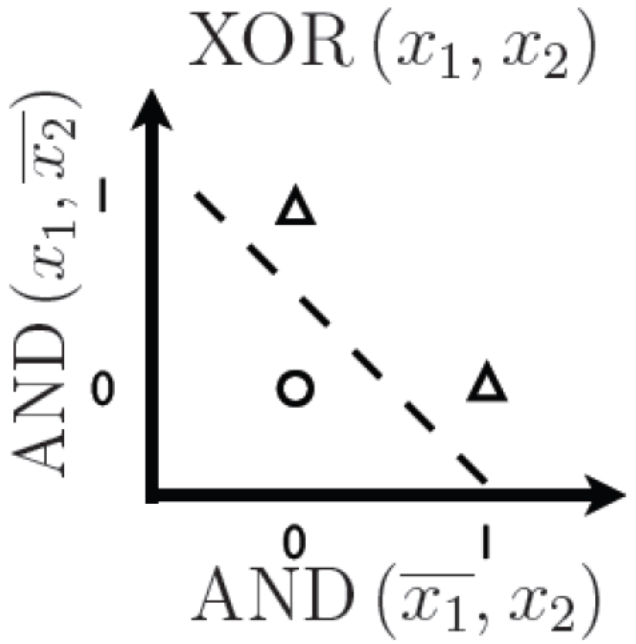
*Images from Hugo Larochelle's course.

Can't solve non-linearly separable problems…



*Images from Hugo Larochelle's course.

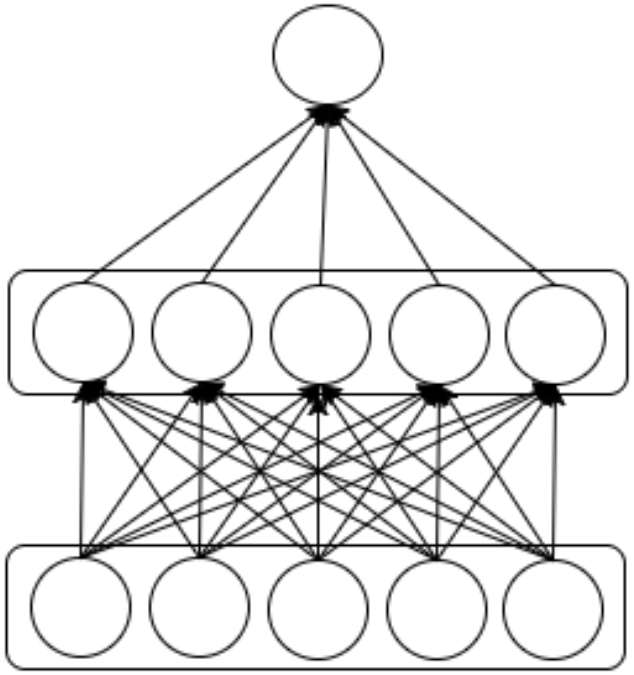
Unless the input is transformed to a better feature space..



*Images from Hugo Larochelle's course.

Can the learning algorithm automatically learn these features?

Neural Networks

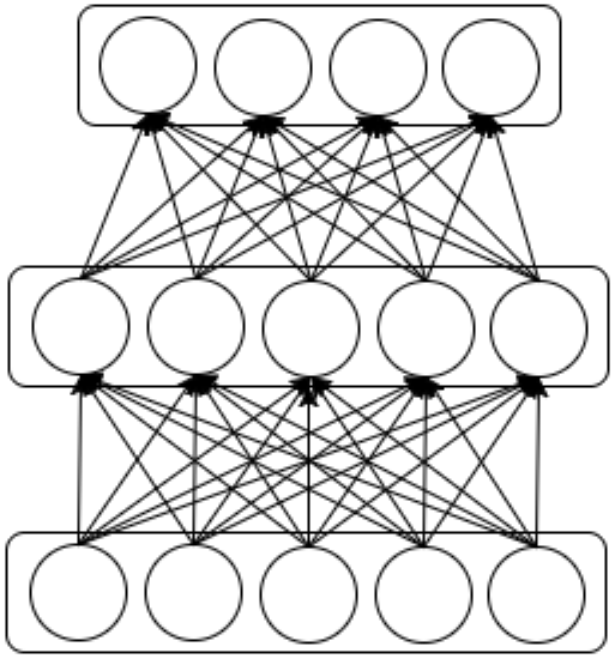


$$h = f(W^{(1)}x + b^{(1)})$$

$$o = W^{(2)}h + b^{(2)}$$

- You need some non-linearity *f*.
- Without *f*, this is still a perceptron!
- *h* – hidden layer.

Neural Networks with multiple outputs



$$h = f(W^{(1)}x + b^{(1)})$$

$$o = W^{(2)}h + b^{(2)}$$

$W^{(2)}$ is a matrix and $b^{(2)}$ is a vector

When you need a probability distribution over n outputs:

$$o = softmax(W^{(2)}h + b^{(2)})$$

$$softmax(a) = \left[\frac{exp(a_1)}{\sum_i exp(a_i)}, ..., \frac{exp(a_n)}{\sum_i exp(a_i)}\right]$$

Training Neural Networks

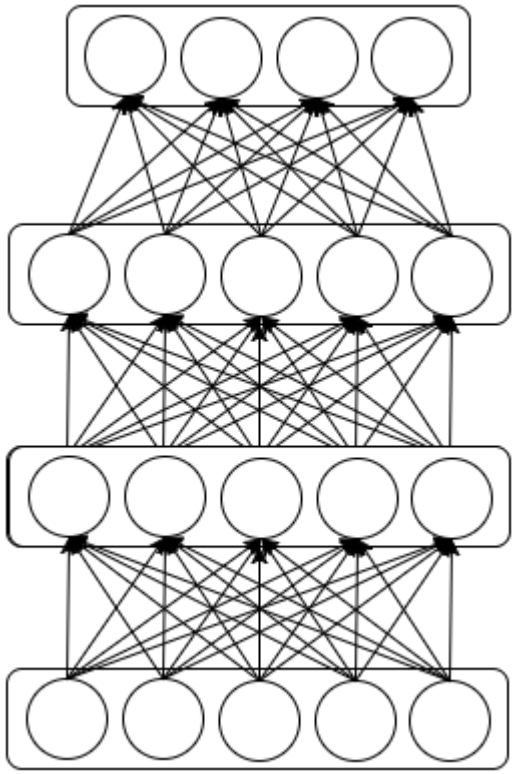
- Learning problem is still same as perceptron learning problem.
- Only the functional form of output is more complicated.
- We can learn the parameters of the neural network $\{W^{(1)}, W^{(2)}, b^{(1)}, b^{(2)}\}$ using gradient descent.
- Can we make use the sequential nature of the neural networks for more efficient computation of gradient?

Backpropagation for Neural Networks

- Algorithm for efficient computation of gradients using chain rule of differentiation.
- Backpropagation is not a *learning* algorithm. We still use gradient descent.

- No more need to derive backprop manually! Theano/Torch/Tensorflow can do it for you!

Deep Neural Networks



- Can have multiple hidden layers.
- More hidden layers, more non-linear the final projection!

$$h^{(1)} = f(W^{(1)}x + b^{(1)})$$

$$h^{(2)} = f(W^{(2)}h^{(1)} + b^{(2)})$$

$$o = softmax(W^{(3)}h^{(2)} + b^{(3)})$$

Language Modeling – An application

- N-gram language modeling: given *n-1* words, predict the *n*-th word.

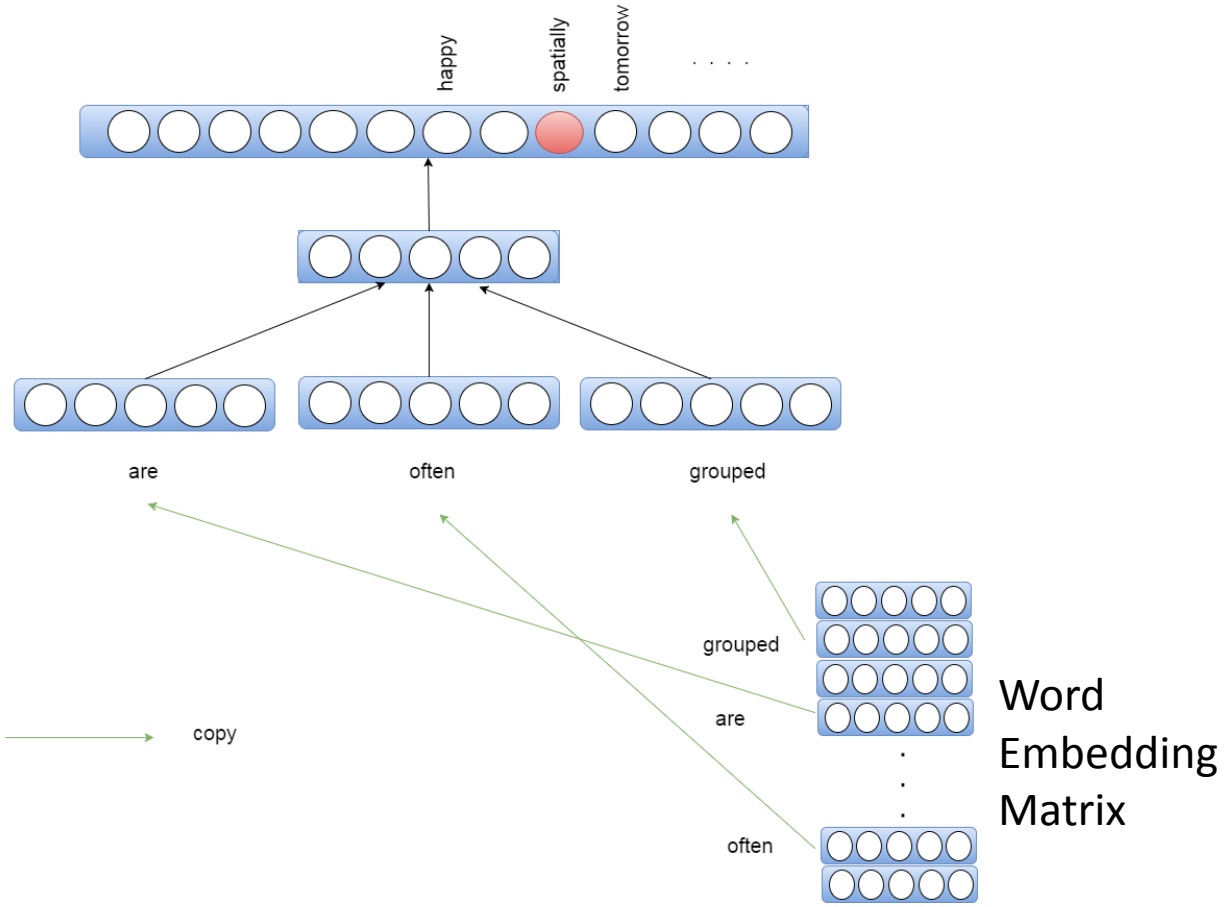
- Example

  Objects are often grouped spatially.

4-gram model will consider ‘are’, ‘often’, ‘grouped’ to predict ‘spatially’.

Traditional n-gram models will use the frequency statistics to compute p(spatially|grouped,often,are).

Neural Language Modeling (Bengio et al., 2001)



- Feed-forward neural network.
- Word embedding matrix W_e is also learnt using backprop+GD.

$$x = [W_e['are']; W_e['often']; W_e['grouped']]$$

$$h = f(W^{(1)}x + b^{(1)})$$

$$o = softmax(W^{(2)}h + b^{(2)})$$

minimize -log $o['spatially']$

Distributed Natural Language Processing

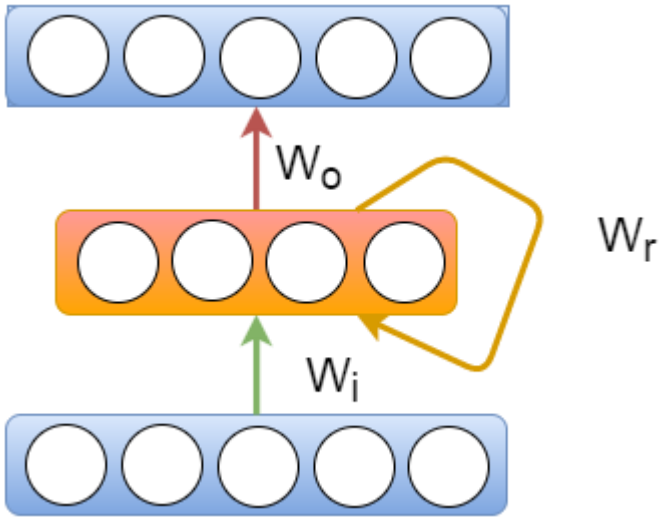
- Neural Networks learn distributed word representation instead of localized word representations.
- Advantages of distributed word representation:
  - Comes with similarity as a by-product.
  - Easy to generalize when compared to localized representations.
  - Can be used to initialize word embedding in other algorithms.

Modeling sequence data

- Feedforward networks ignore the sequence information.
- We managed to include some sequence information in neural language model by considering previous *n-1* words.

- Can we implicitly add this sequence information in the network architecture?

Recurrent Neural Networks



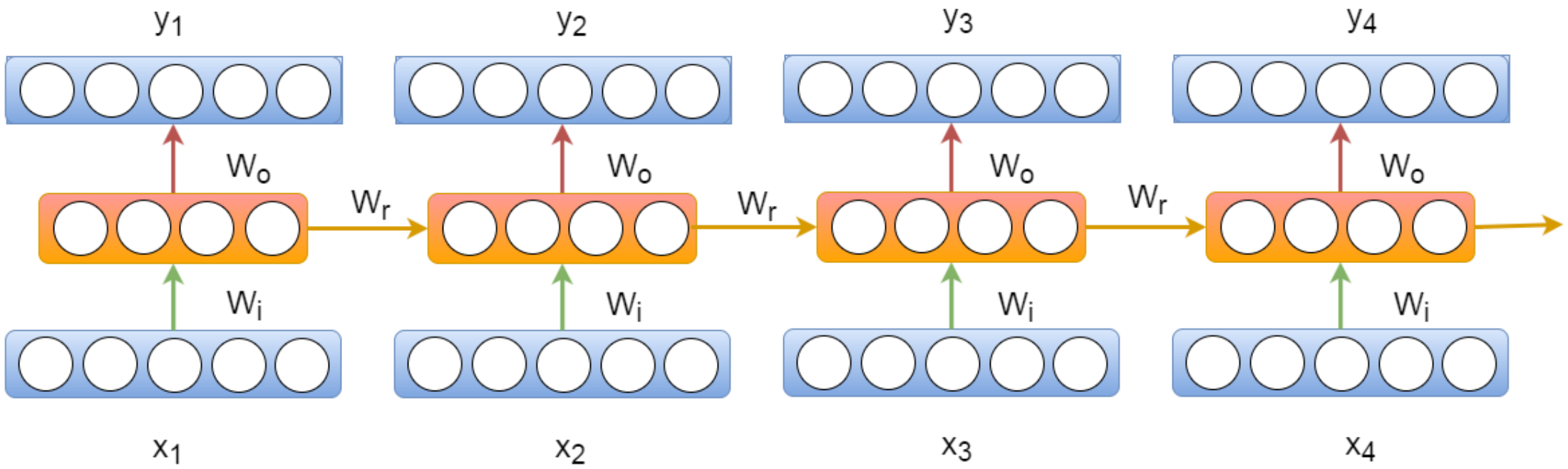
$$h_t = f(W_i x_t + W_r h_{t-1} + b)$$

$$o_t = softmax(W_o h_t + c)$$

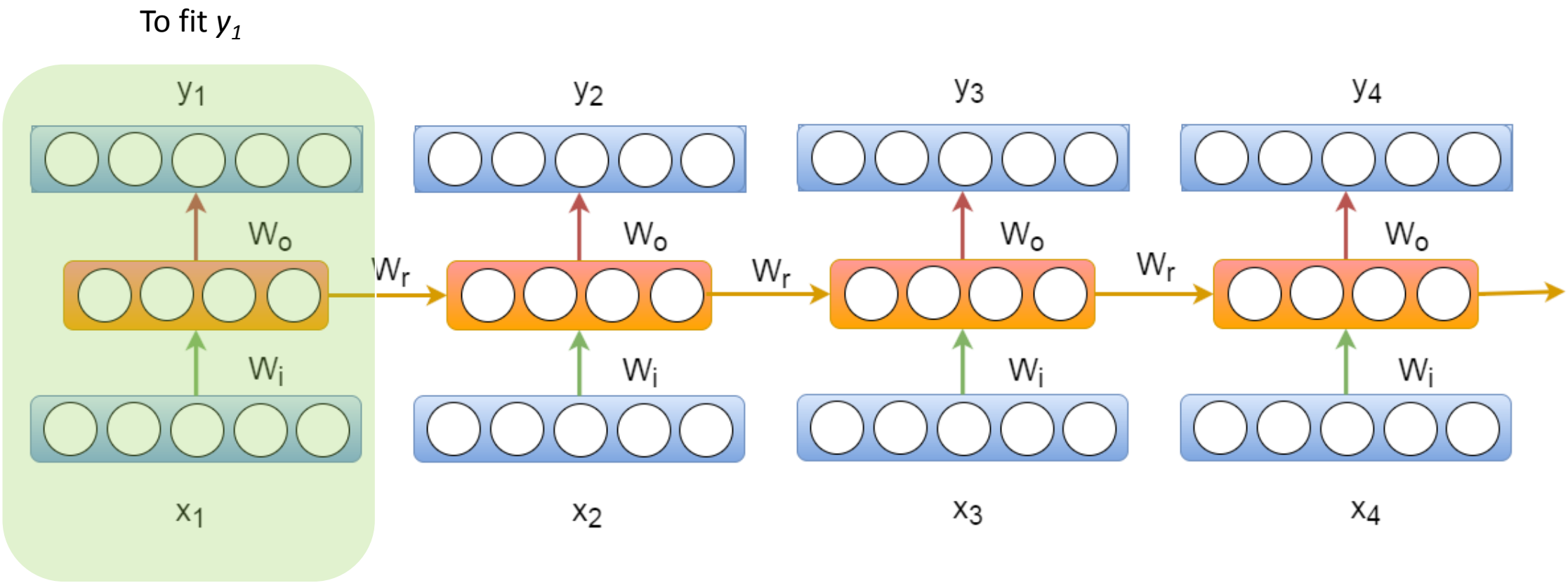
Input: $x_1, x_2, x_3, ..., x_n$
Output: $y_1, y_2, y_3, ..., y_n$

h_0 can be initialized to a zero vector or learned as a parameter.

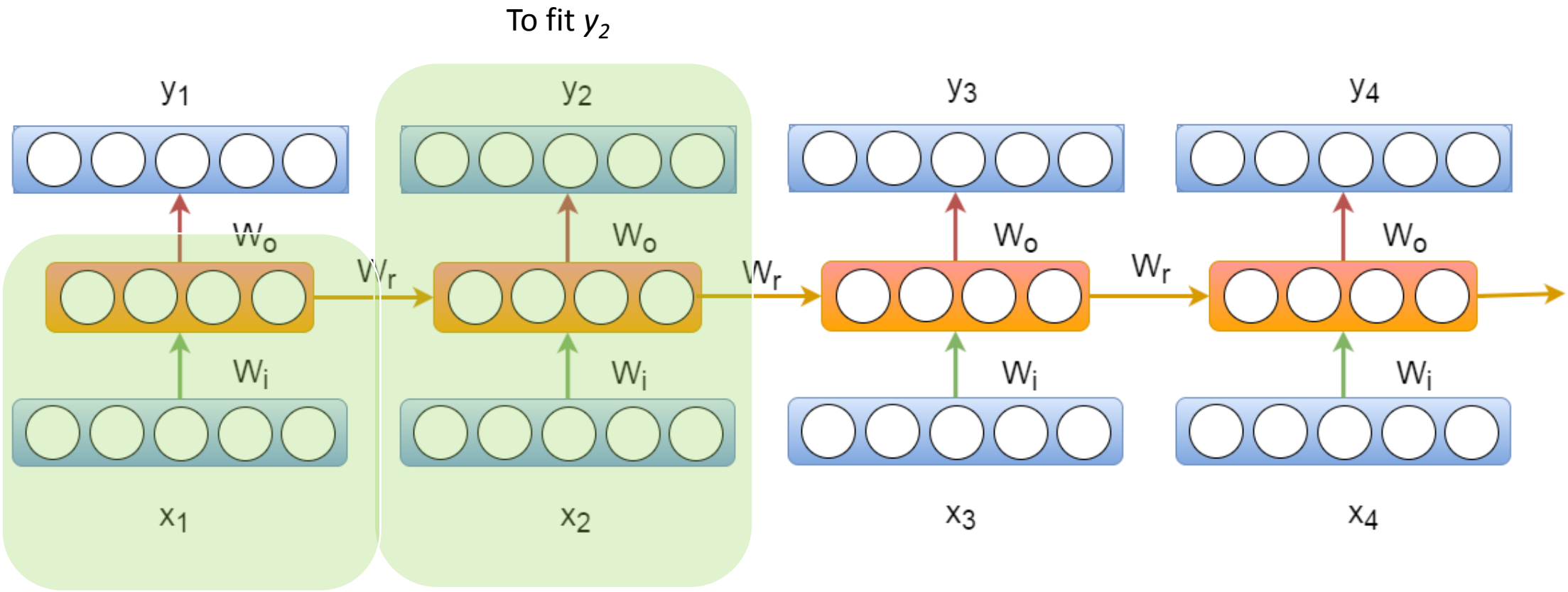
Recurrent Neural Networks



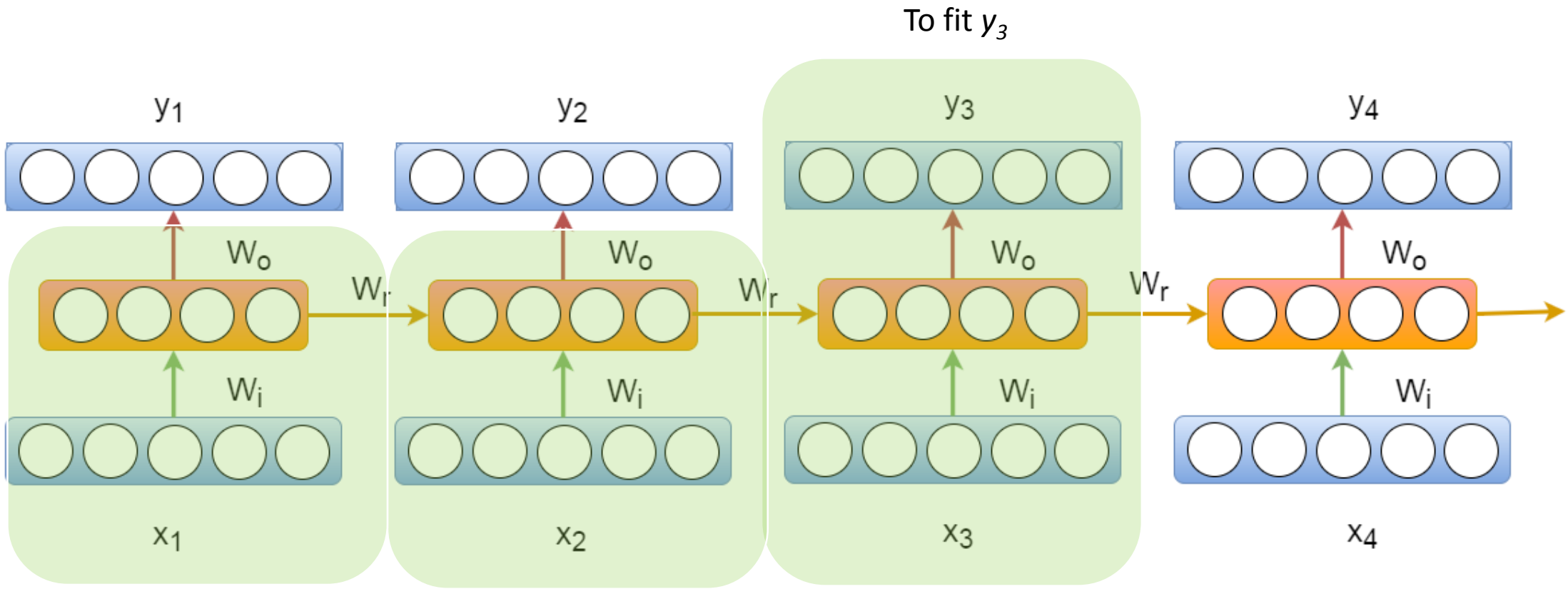
Backpropagation Through Time (BPTT)



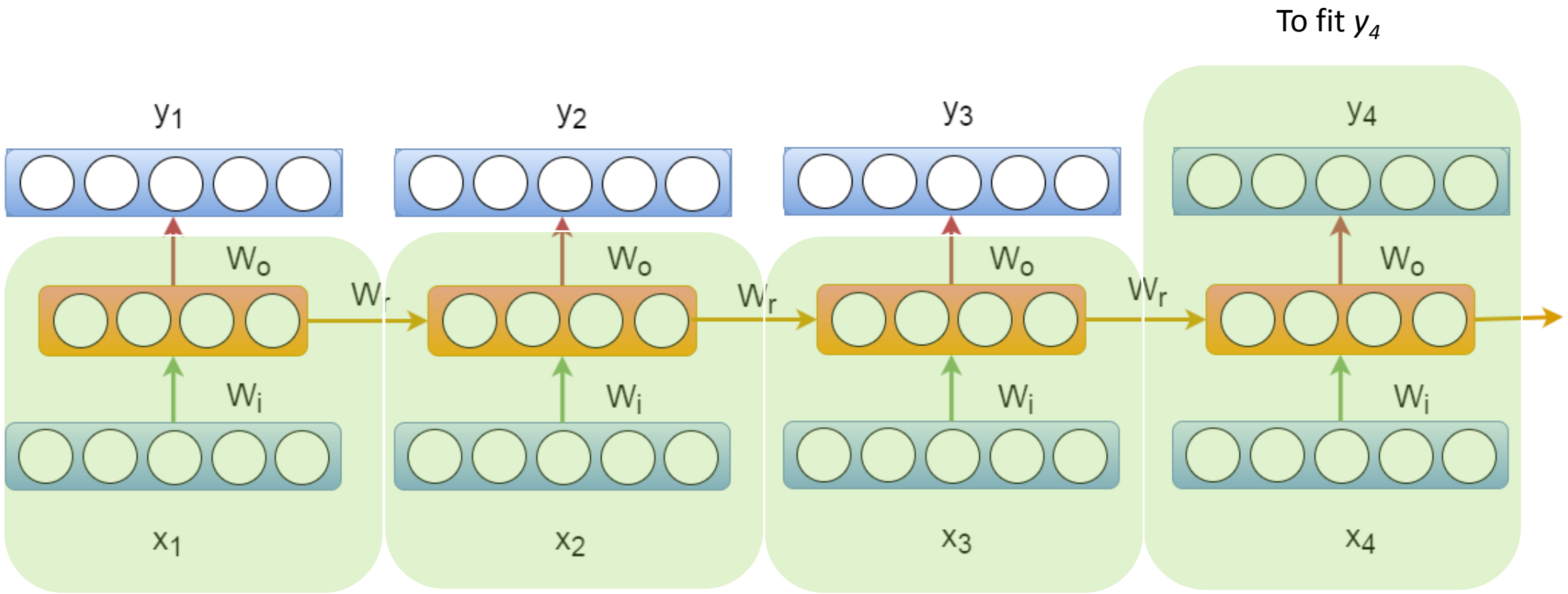
Backpropagation Through Time (BPTT)



Backpropagation Through Time (BPTT)



Backpropagation Through Time (BPTT)

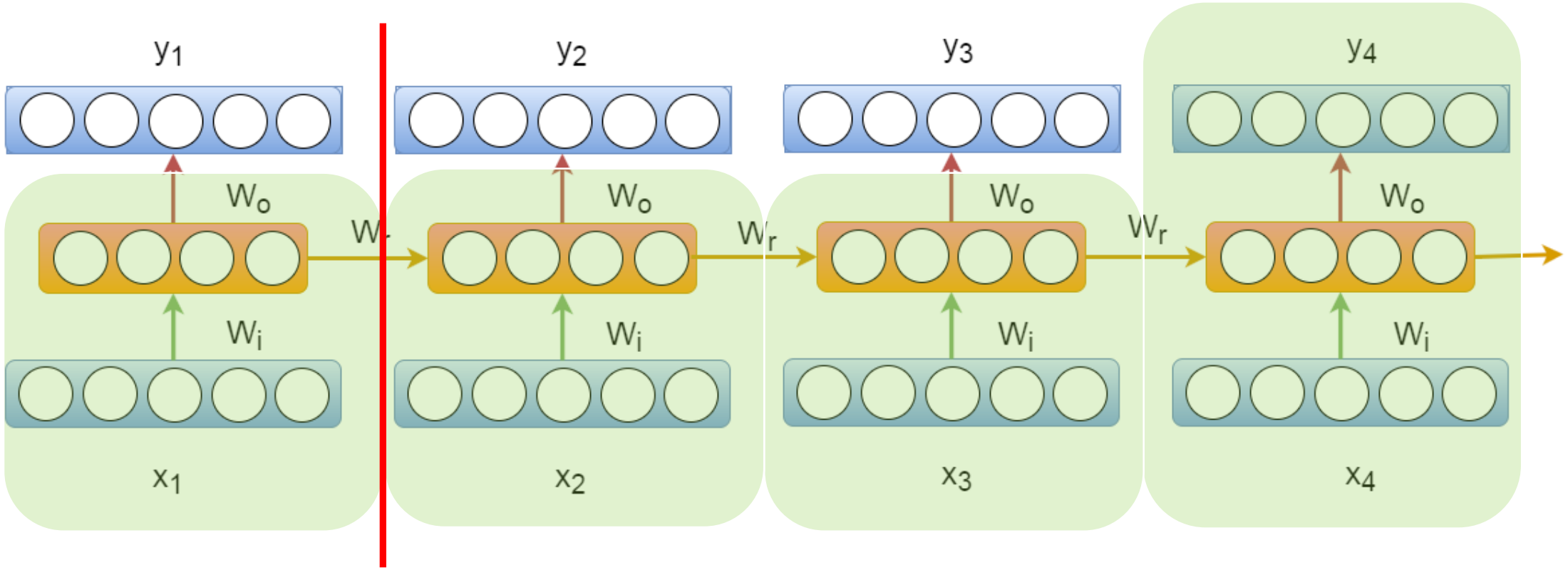


Computationally expensive for longer sequences !

Truncated Backpropagation Through Time (T-BPTT)

Truncated after 2 steps.

To fit y_4

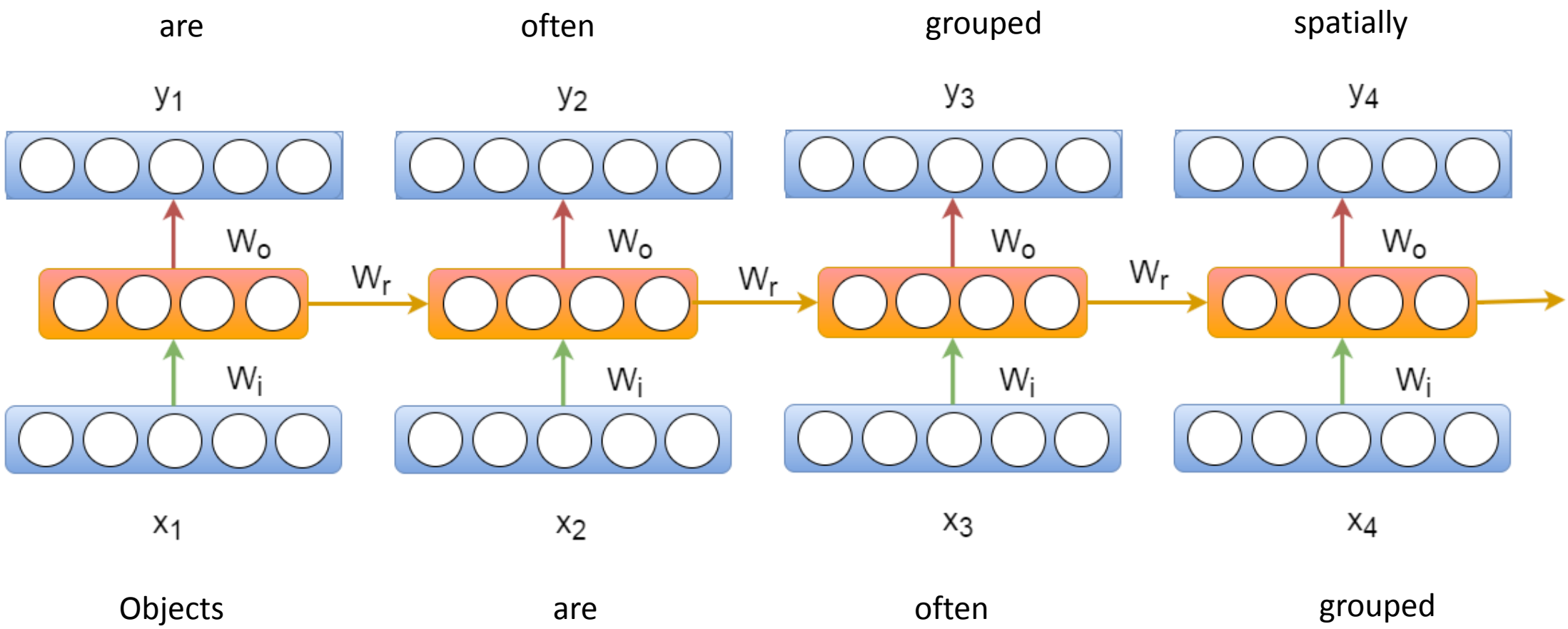


RNN Language Model (Mikolov et al., 2010)

Language modeling as sequential prediction problem.

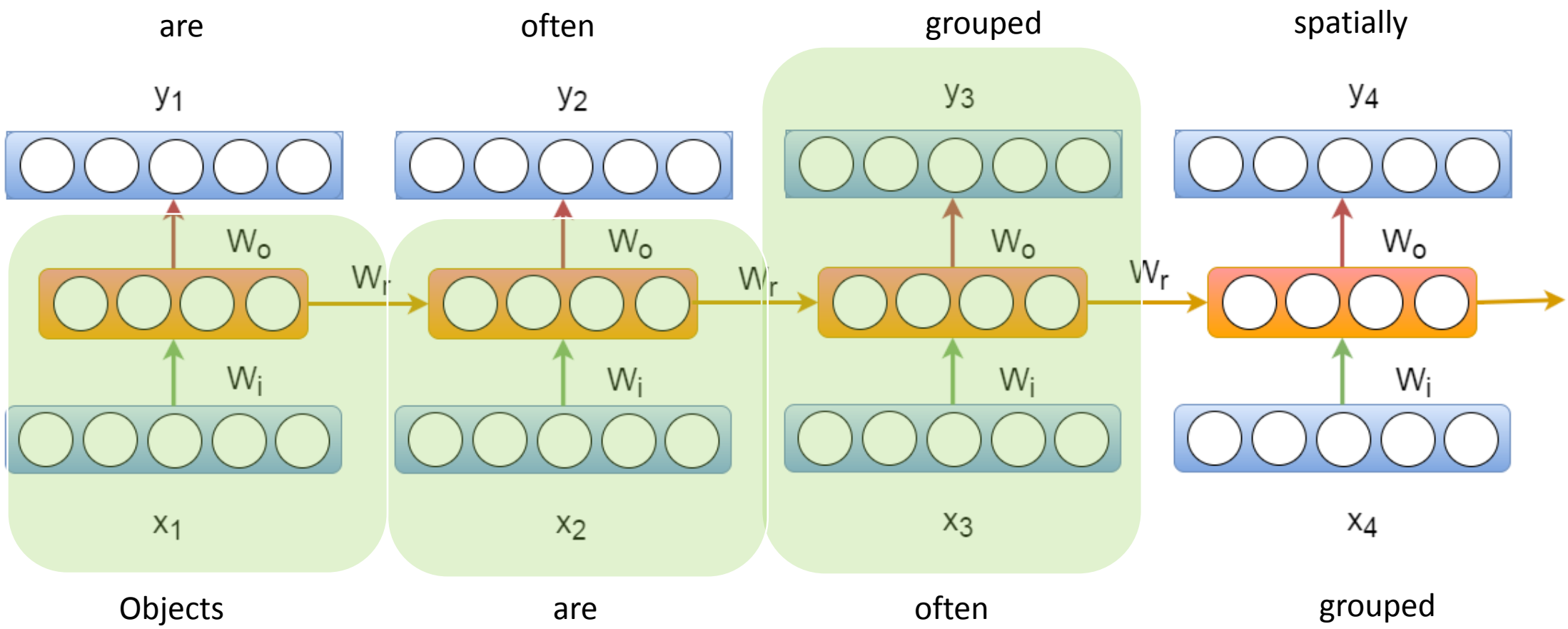
I/P :	Objects	are	often	grouped	spatially	.
O/P:	are	often	grouped	spatially	.	<EOS>

RNN Language model



RNN Language model

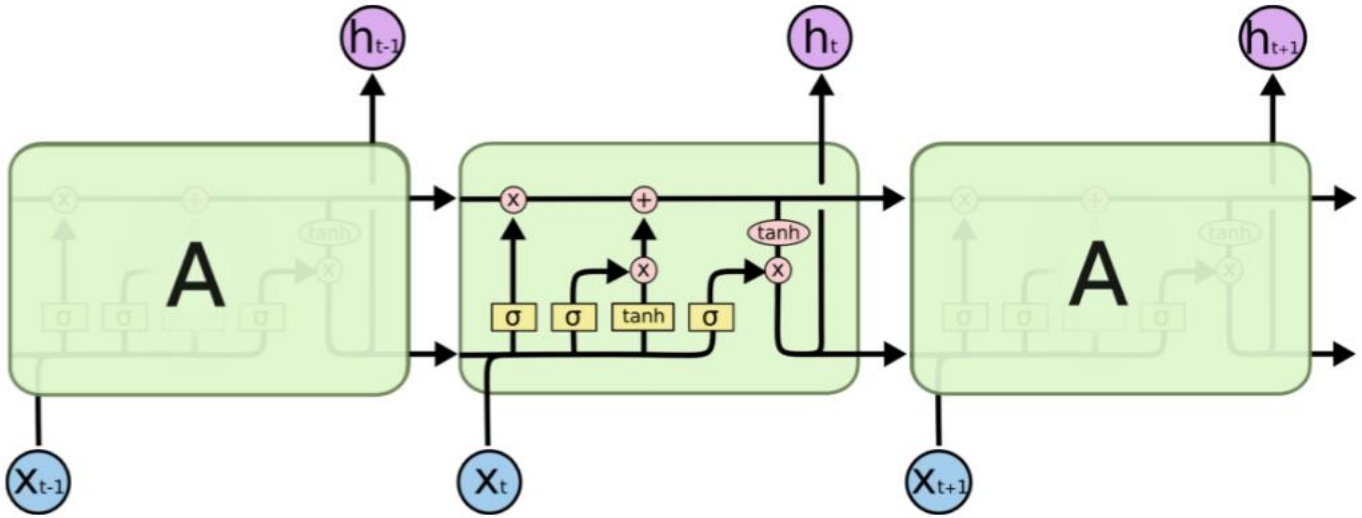
Models p(grouped|often,are,objects)



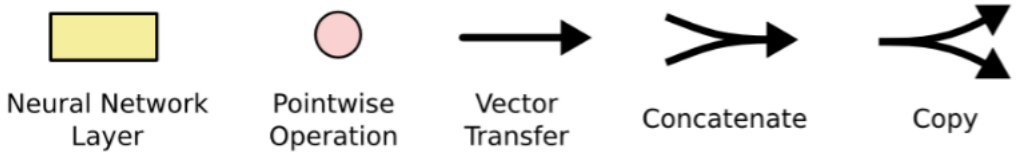
Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997)

- LSTM is a variant of RNN that is good at modeling long-term dependencies.
- RNN uses multiplication to overwrite the hidden state while LSTM uses addition (better gradient flow!).

Long Short Term Memory (LSTM)

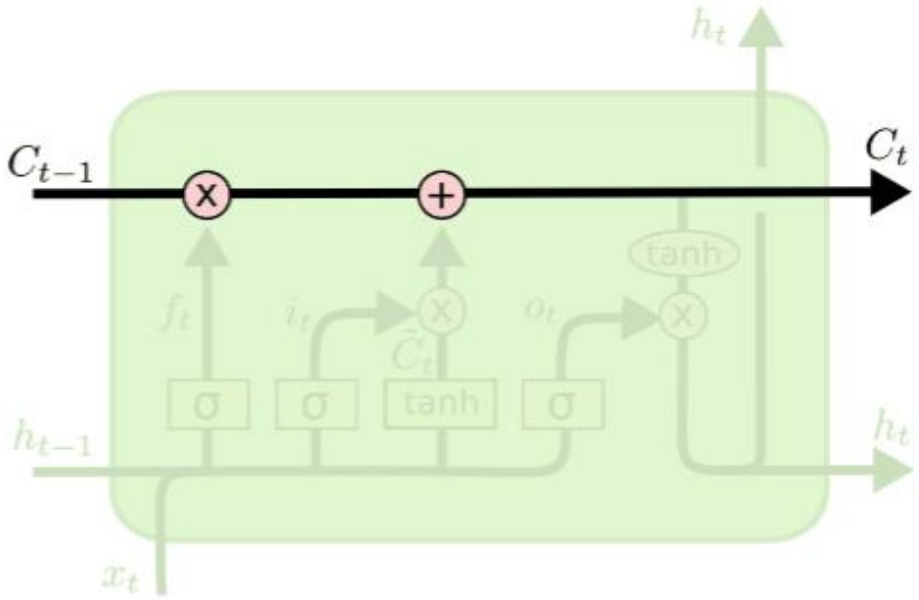


The repeating module in an LSTM contains four interacting layers.



*Pictures from Chris Olah

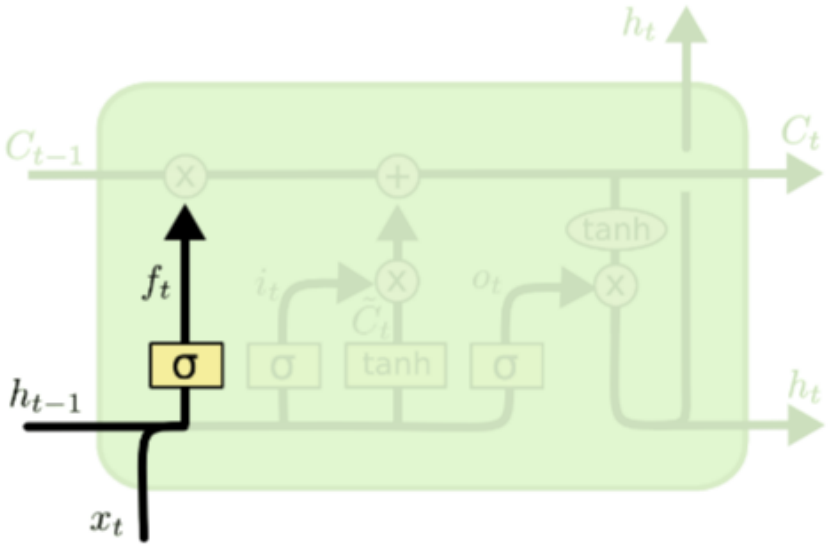
Long Short Term Memory (LSTM)



Additive context

*Pictures from Chris Olah

Long Short Term Memory (LSTM)

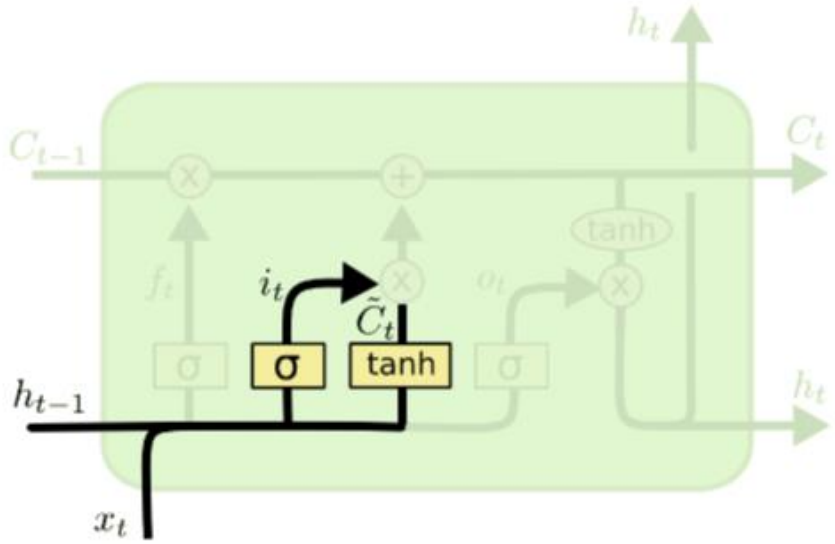


Forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

*Pictures from Chris Olah

Long Short Term Memory (LSTM)



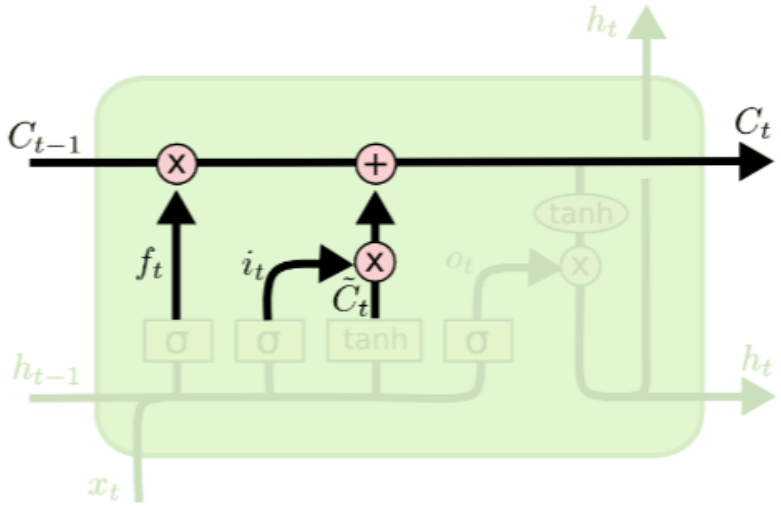
Input gate

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

*Pictures from Chris Olah

Long Short Term Memory (LSTM)

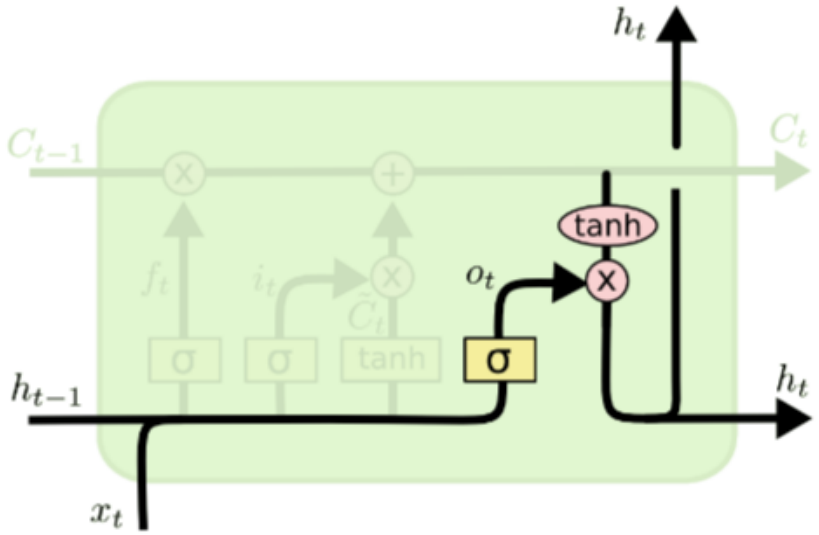


Additive context

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

*Pictures from Chris Olah

Long Short Term Memory (LSTM)



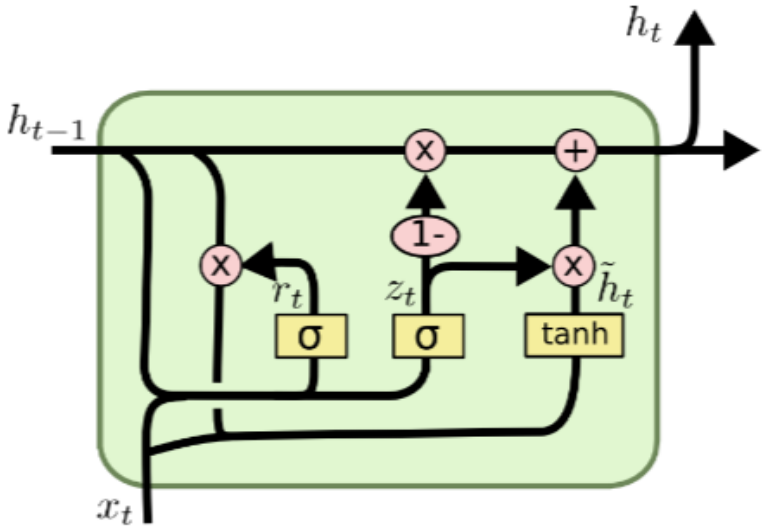
Output gate

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

*Pictures from Chris Olah

Gated Recurrent Units (GRU) (Cho et al., 2014)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

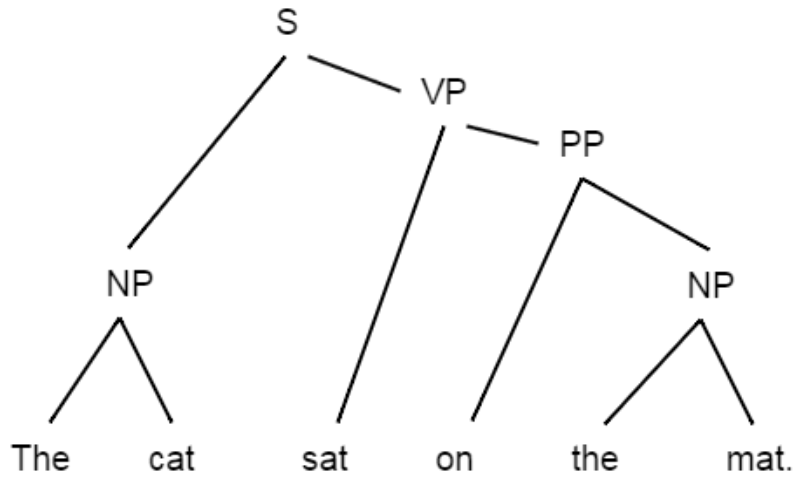
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

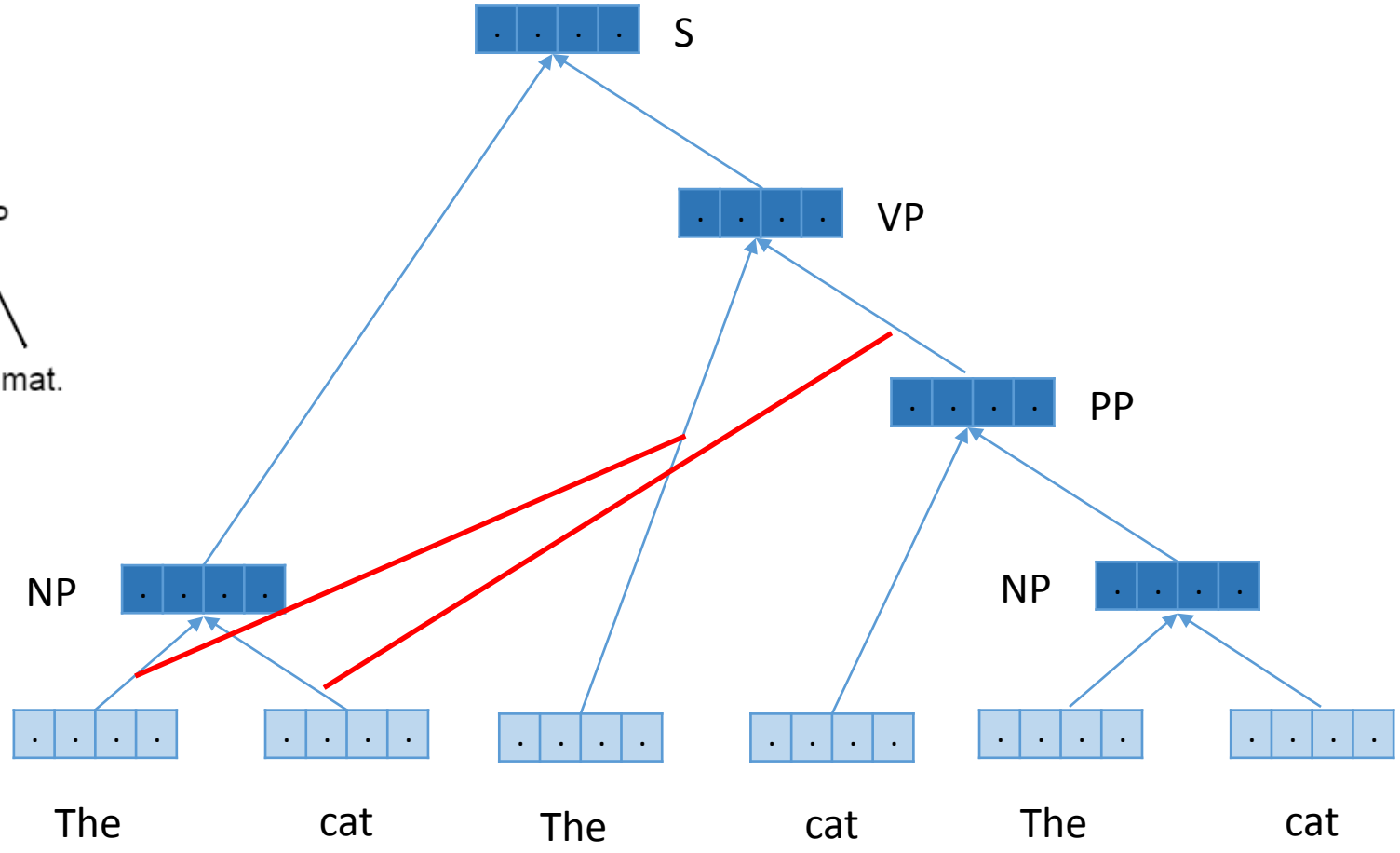
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

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

*Pictures from Chris Olah

Recursive Neural Networks (Pollack., 1990)

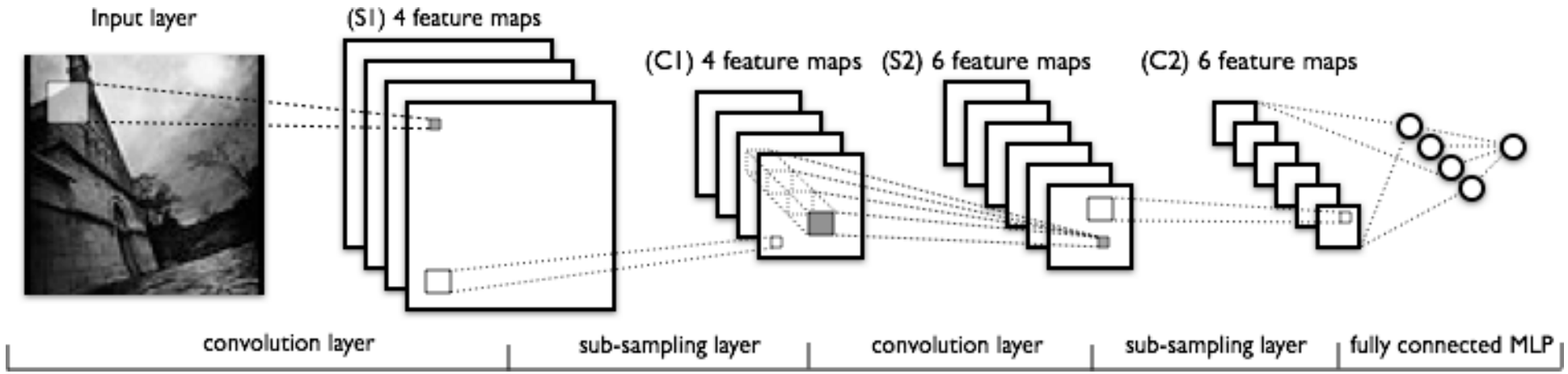




Shared weights

Convolutional Networks

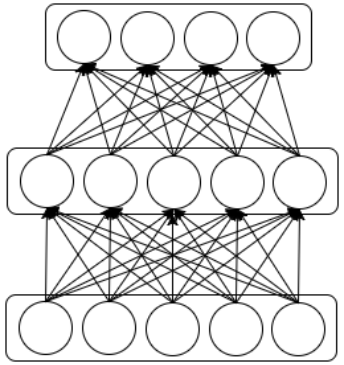
- Standard network architecture for image representation.



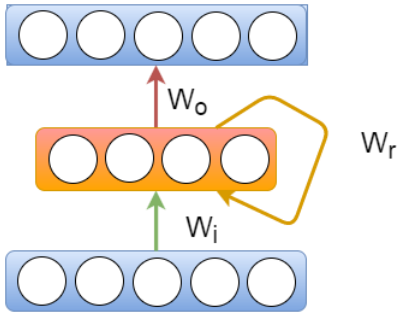
State of the art performance in object recognition, object detection, image segmentation…

Matching data with architecture

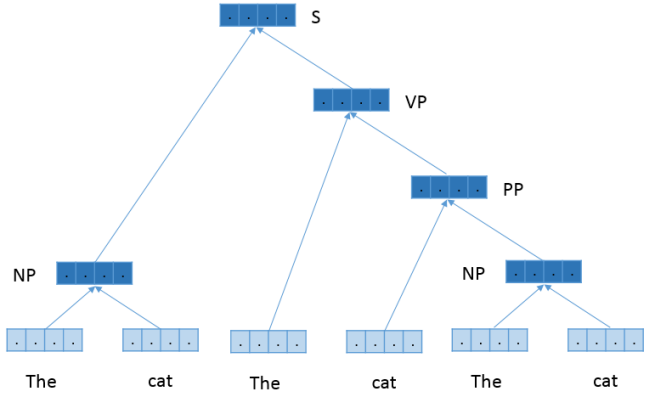
Bag-of-words like data



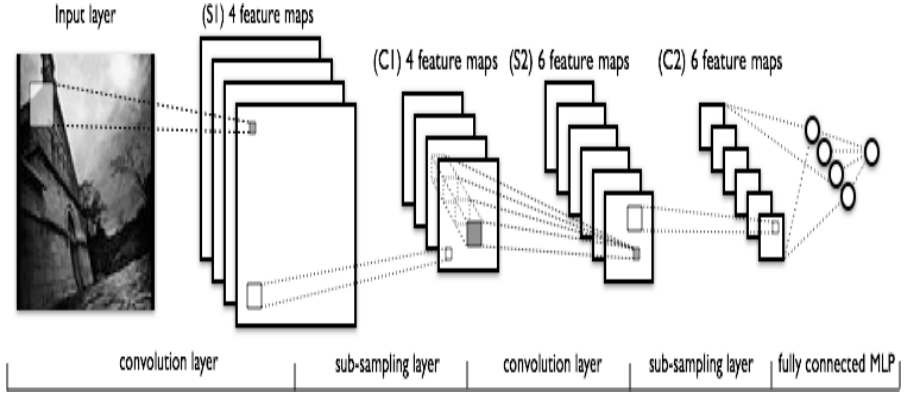
Sequence data



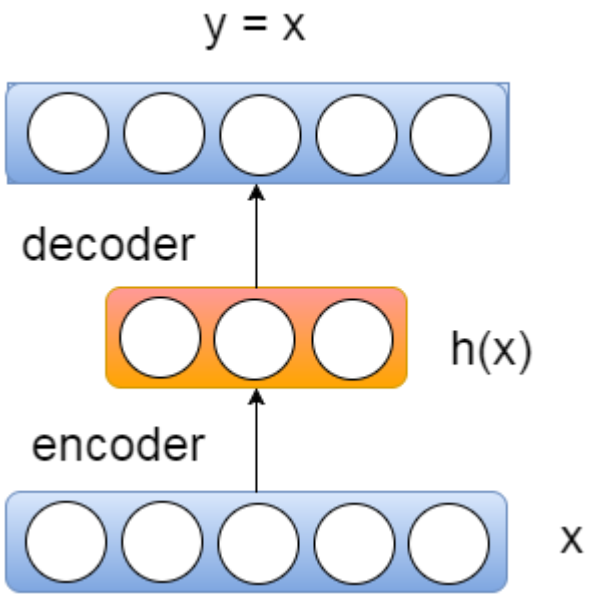
Tree structured data



Images

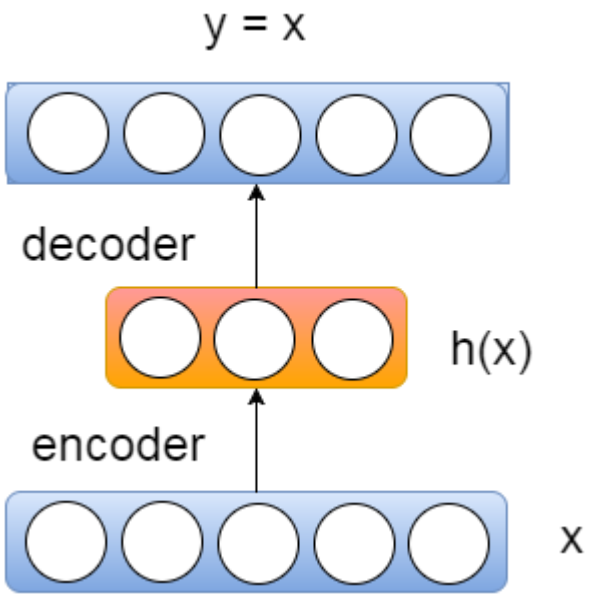


Autoencoders



- Consists of two modules: encoder and decoder.
- Output is equal to the input.

Autoencoders



$$h(x) = f(Wx + b)$$

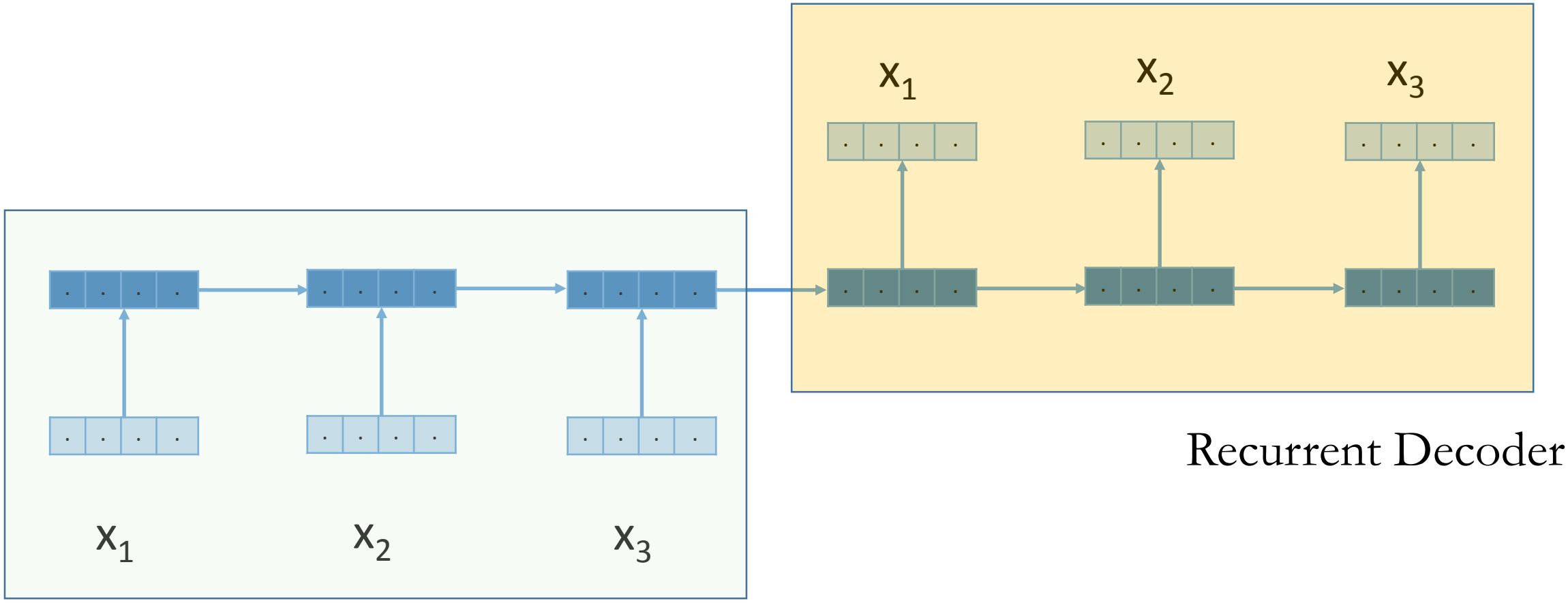
$$o(x) = Vh(x) + c$$

Given $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ minimize

$$\mathcal{J}(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (o(x^{(i)}) - x^{(i)})^2$$

This is for bag-of-words like data.

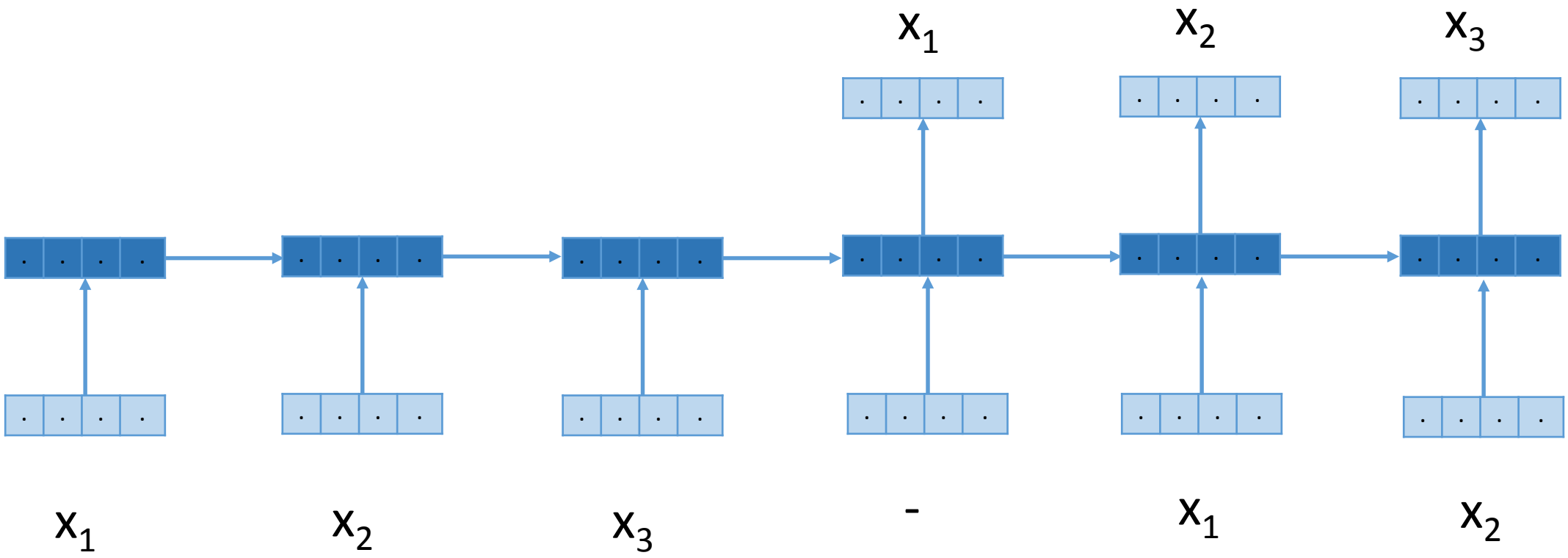
Recurrent Autoencoder



Recurrent Decoder

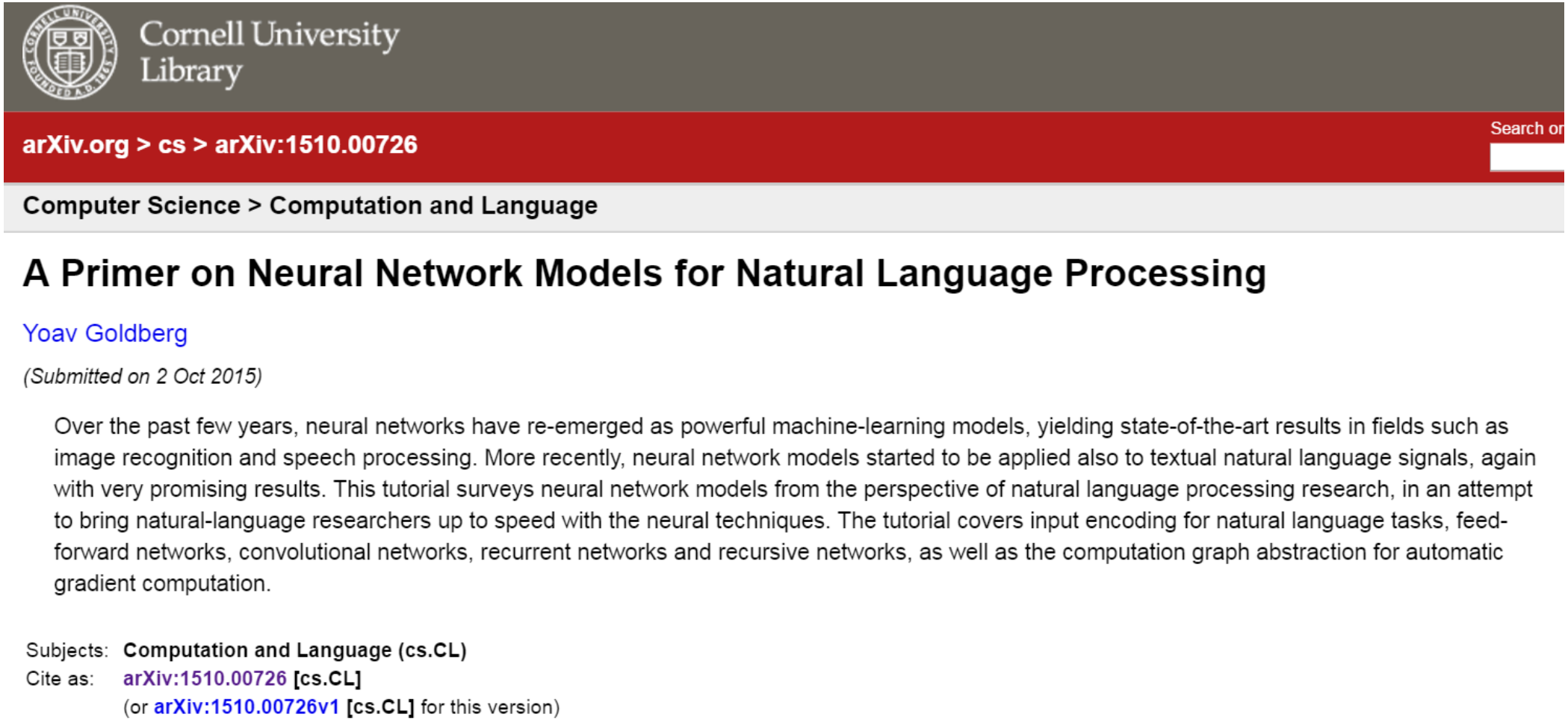
Recurrent Encoder

Recurrent Autoencoder (v2)



You can also imagine Recursive Autoencoders, Convolutional Autoencoders…

For a quick introduction:



Cornell University Library

arXiv.org > cs > arXiv:1510.00726

Search or

Computer Science > Computation and Language

A Primer on Neural Network Models for Natural Language Processing

Yoav Goldberg

(Submitted on 2 Oct 2015)

Over the past few years, neural networks have re-emerged as powerful machine-learning models, yielding state-of-the-art results in fields such as image recognition and speech processing. More recently, neural network models started to be applied also to textual natural language signals, again with very promising results. This tutorial surveys neural network models from the perspective of natural language processing research, in an attempt to bring natural-language researchers up to speed with the neural techniques. The tutorial covers input encoding for natural language tasks, feed-forward networks, convolutional networks, recurrent networks and recursive networks, as well as the computation graph abstraction for automatic gradient computation.

Subjects: **Computation and Language (cs.CL)**
Cite as: **arXiv:1510.00726 [cs.CL]**
(or **arXiv:1510.00726v1 [cs.CL]** for this version)

For a detailed introduction:

Exercises Lecture Slides

The Deep Learning textbook is a resource intended to help students and practitioners enter the field of machine learning in general and deep learning in particular. The online version of the book is now complete and will remain available online for free. The print version will be available for sale soon. For up to date announcements, join our mailing list.

Citing the book

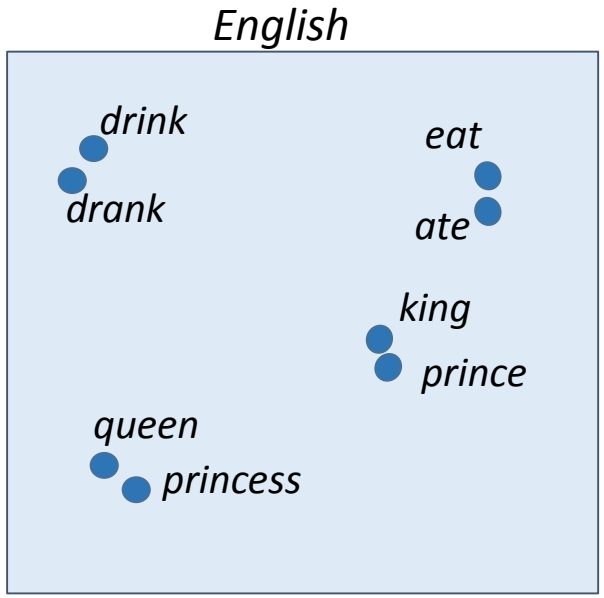
To cite this book, please use this bibtex entry:

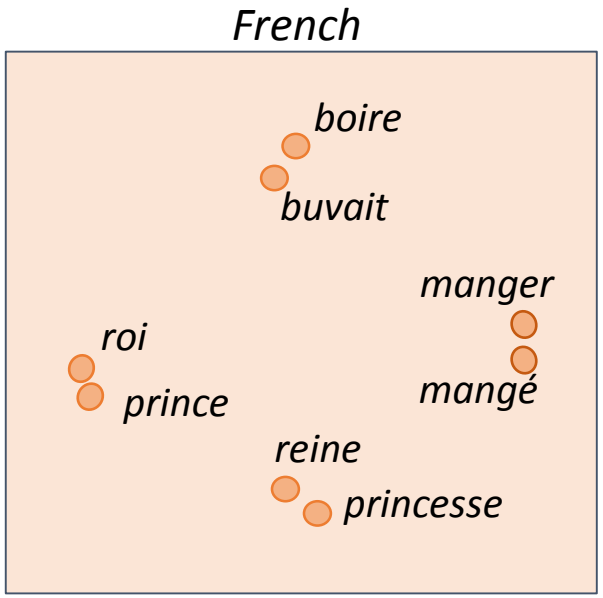
```
@unpublished{Goodfellow-et-al-2016-Book,
    title={Deep Learning},
    author={Ian Goodfellow Yoshua Bengio and Aaron Courville},
    note={Book in preparation for MIT Press},
    url={http://www.deeplearningbook.org},
    year={2016}
}
```

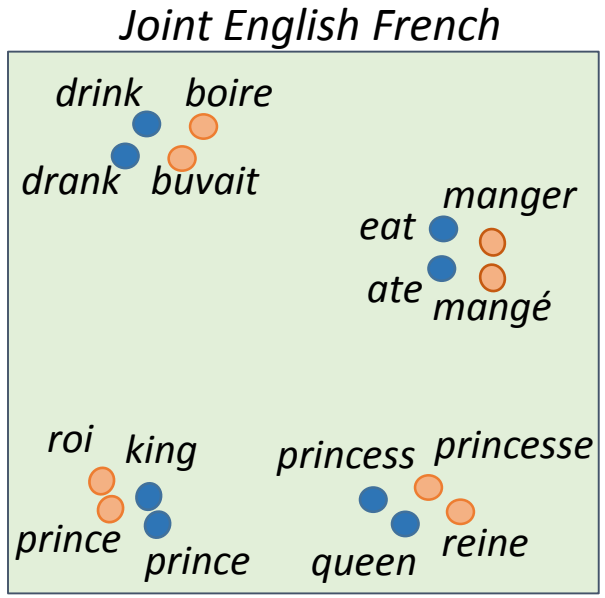
FAQ

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
5. Summary and research directions







Monolingual Word Representations
(capture syntactic and semantic similarities between words)

Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Lets start by defining the goal of learning multilingual word representations

First lets try to understand how do we learn monolingual word representations

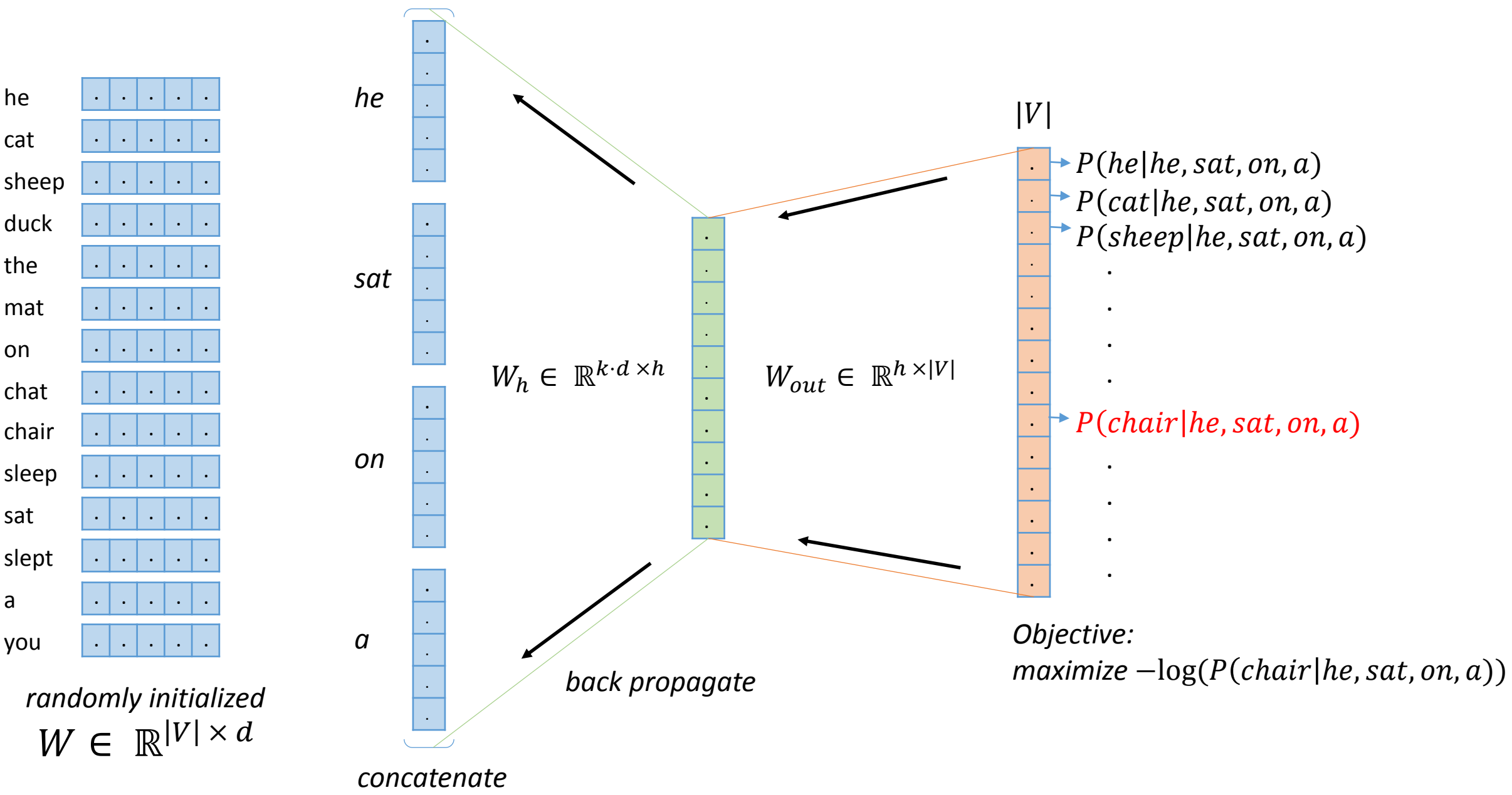
Consider this Task: Predict n-th word given previous n-1 words

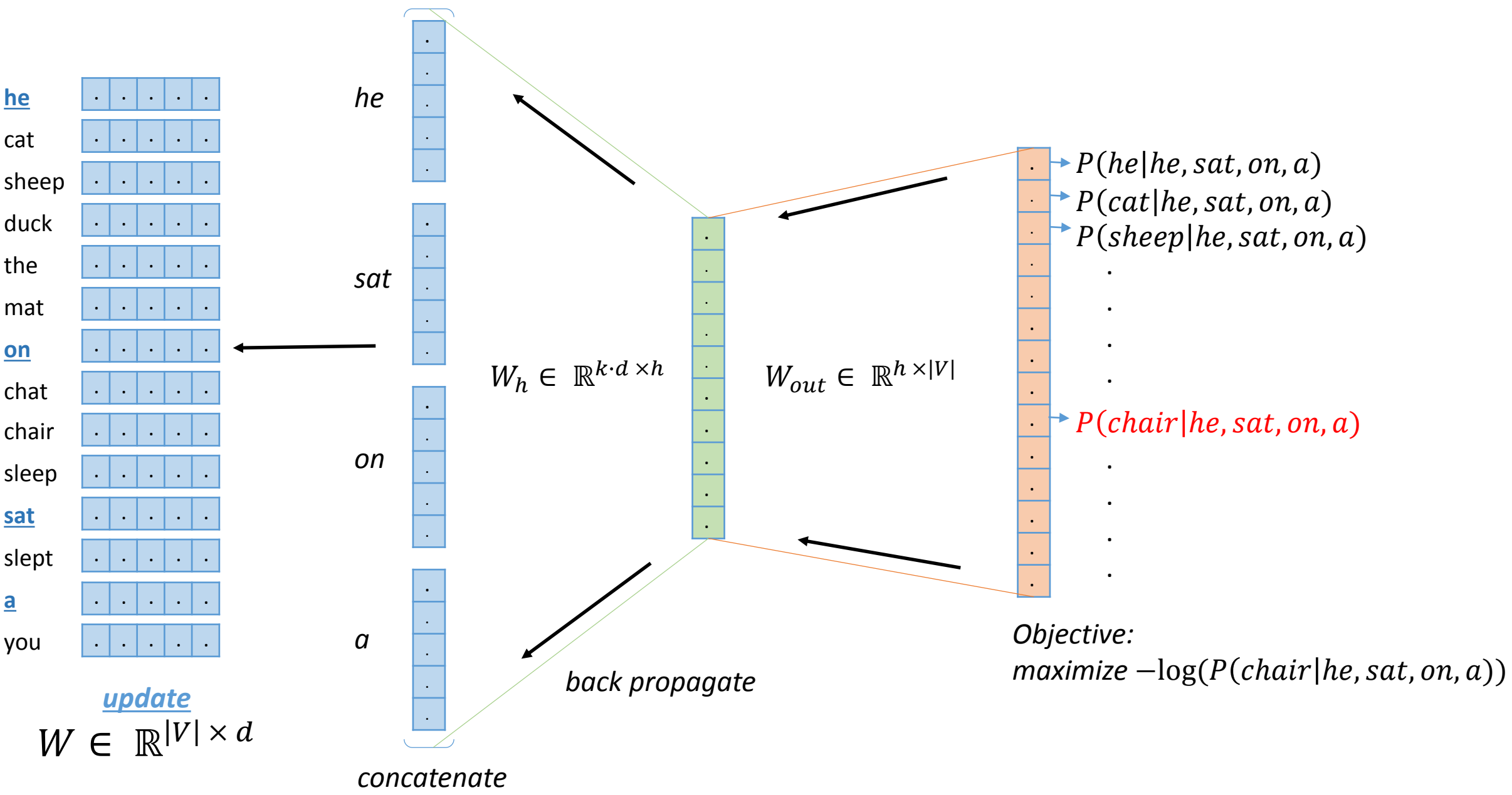
Example: he sat on a chair

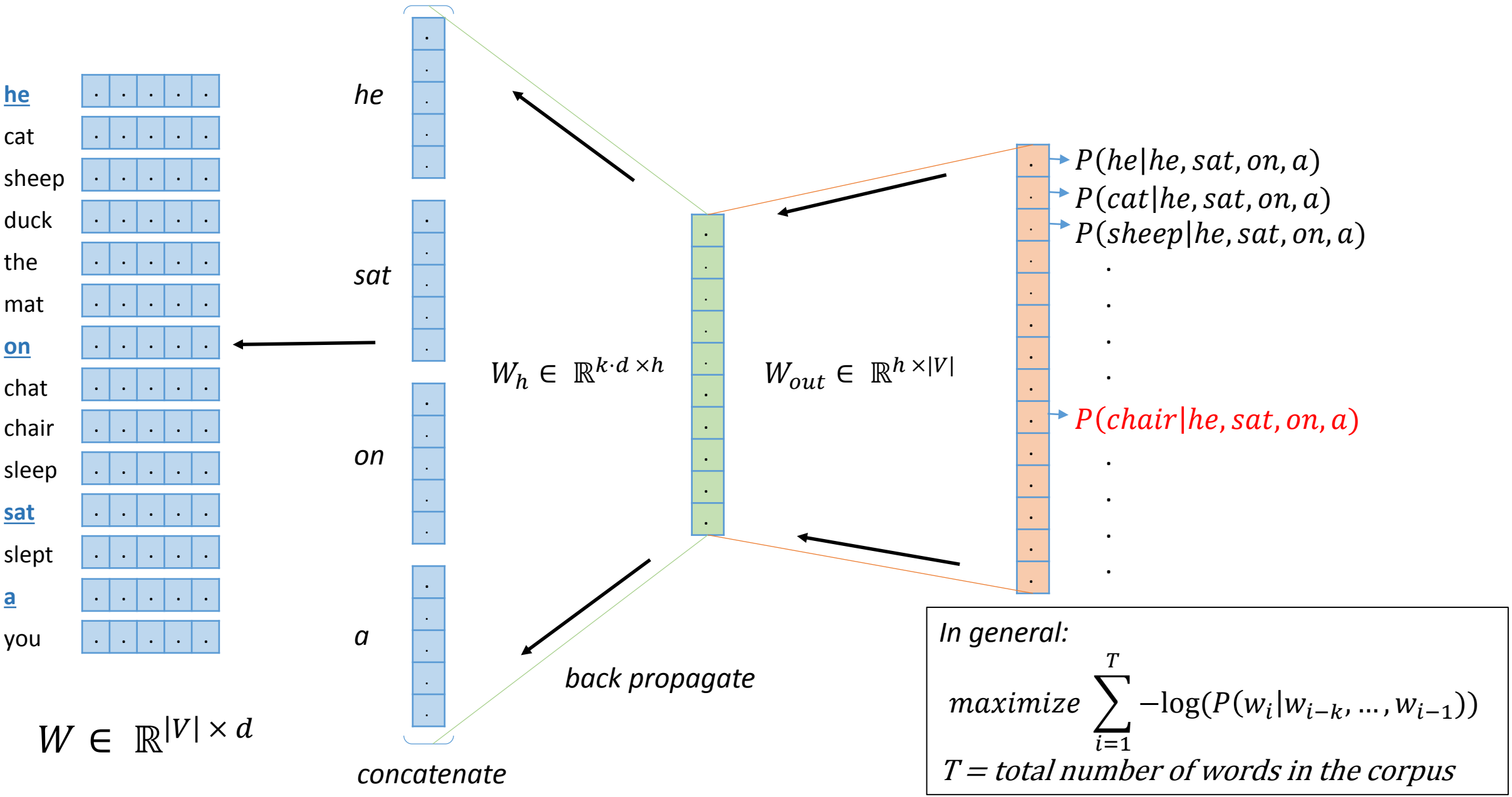
Training data: All n-word windows in your corpus

Now, lets try to answer these two questions:

- *How do you model this task?*
- *What is the connection between this task and learning word representations?*


Training instance: he sat on a chair
he
cat
sheep
duck
the
mat
on
chat
chair
sleep
sat
slept
a
you
randomly initialized
$W \in \mathbb{R}^{|V| \times d}$
he
sat
on
a
concatenate
$W_h \in \mathbb{R}^{k \cdot d \times h}$
$W_{out} \in \mathbb{R}^{h \times |V|}$
back propagate
$|V|$
$P(he|he, sat, on, a)$
$P(cat|he, sat, on, a)$
$P(sheep|he, sat, on, a)$
$P(chair|he, sat, on, a)$
Objective:
maximize $-\log(P(chair|he, sat, on, a))$


Training instance: he sat on a chair
he
cat
sheep
duck
the
mat
on
chat
chair
sleep
sat
slept
a
you
update
$W \in \mathbb{R}^{|V| \times d}$
he
sat
on
a
concatenate
$W_h \in \mathbb{R}^{k \cdot d \times h}$
$W_{out} \in \mathbb{R}^{h \times |V|}$
back propagate
$P(he|he, sat, on, a)$
$P(cat|he, sat, on, a)$
$P(sheep|he, sat, on, a)$
$P(chair|he, sat, on, a)$
Objective:
maximize $-\log(P(chair|he, sat, on, a))$


Training instance: he sat on a chair
he
cat
sheep
duck
the
mat
on
chat
chair
sleep
sat
slept
a
you
$W \in \mathbb{R}^{|V| \times d}$
he
sat
on
a
concatenate
$W_h \in \mathbb{R}^{k \cdot d \times h}$
$W_{out} \in \mathbb{R}^{h \times |V|}$
$P(he|he, sat, on, a)$
$P(cat|he, sat, on, a)$
$P(sheep|he, sat, on, a)$
$P(chair|he, sat, on, a)$
back propagate
In general:
$maximize \sum_{i=1}^{T} -\log(P(w_i|w_{i-k}, \ldots, w_{i-1}))$
T = total number of words in the corpus

How does this result in meaningful word representations?

Intuition: similar words appear in similar contexts

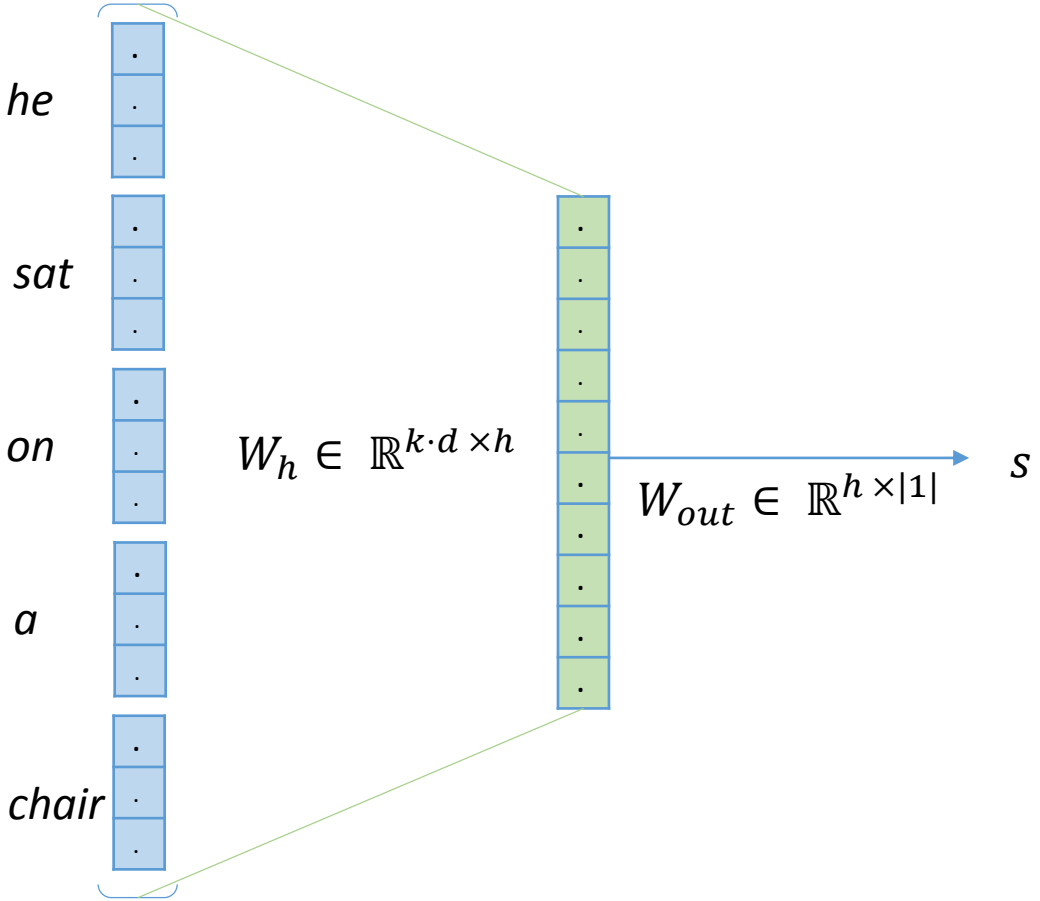
he sat on a chair

he sits on a chair

To predict chair in both cases the model should learn to make the representations of “sits” and “sat” similar

Alternate formulation … … …

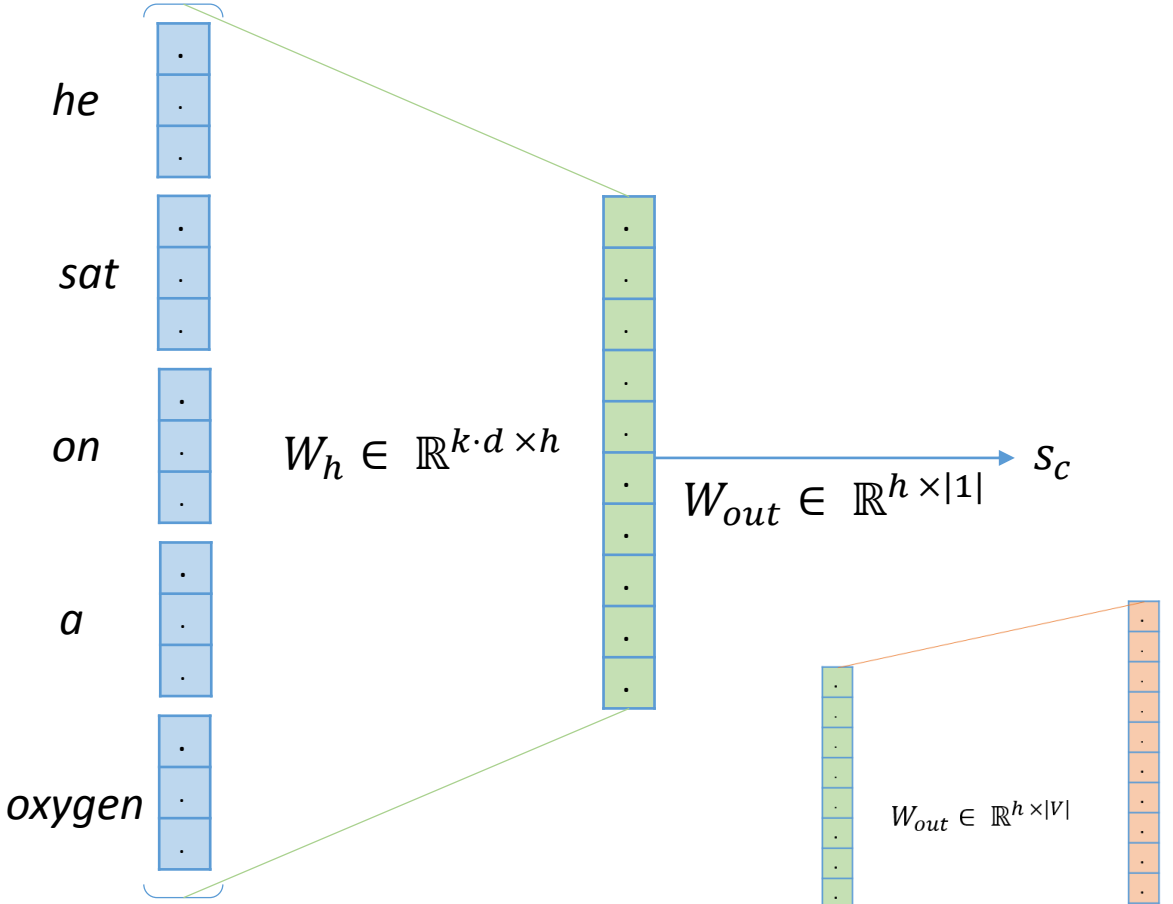
Positive: he sat on a chair



$$maximize\ (0, 1 - s + s_c)$$

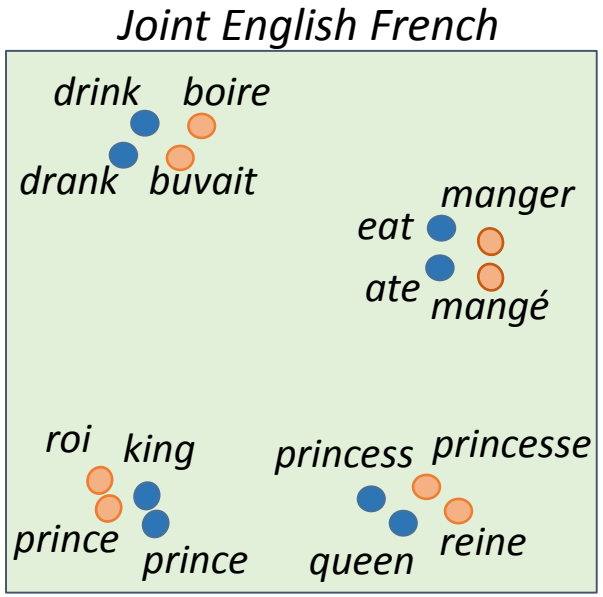
back propagate and update word representations

Negative: he sat on a oxygen



Advantage: does not require this expensive matrix multiplication

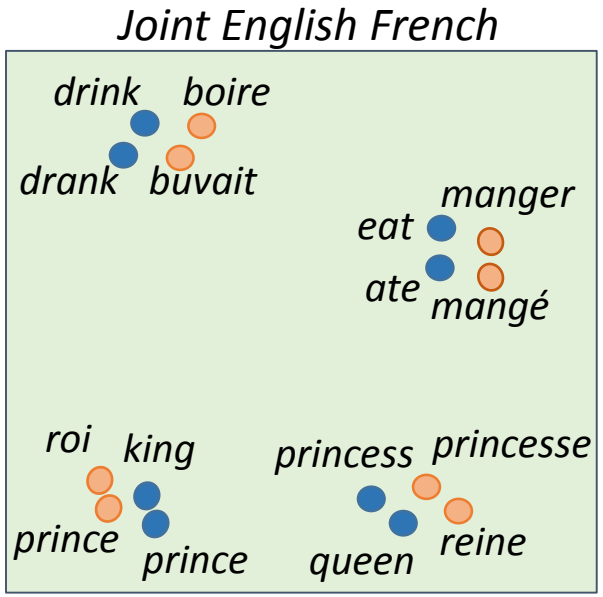
Coming back to learning (multi)bilingual representations



Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

Two paradigms:
- *Offline Bilingual Alignment*
- *Joint training for Bilingual Alignment*



Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

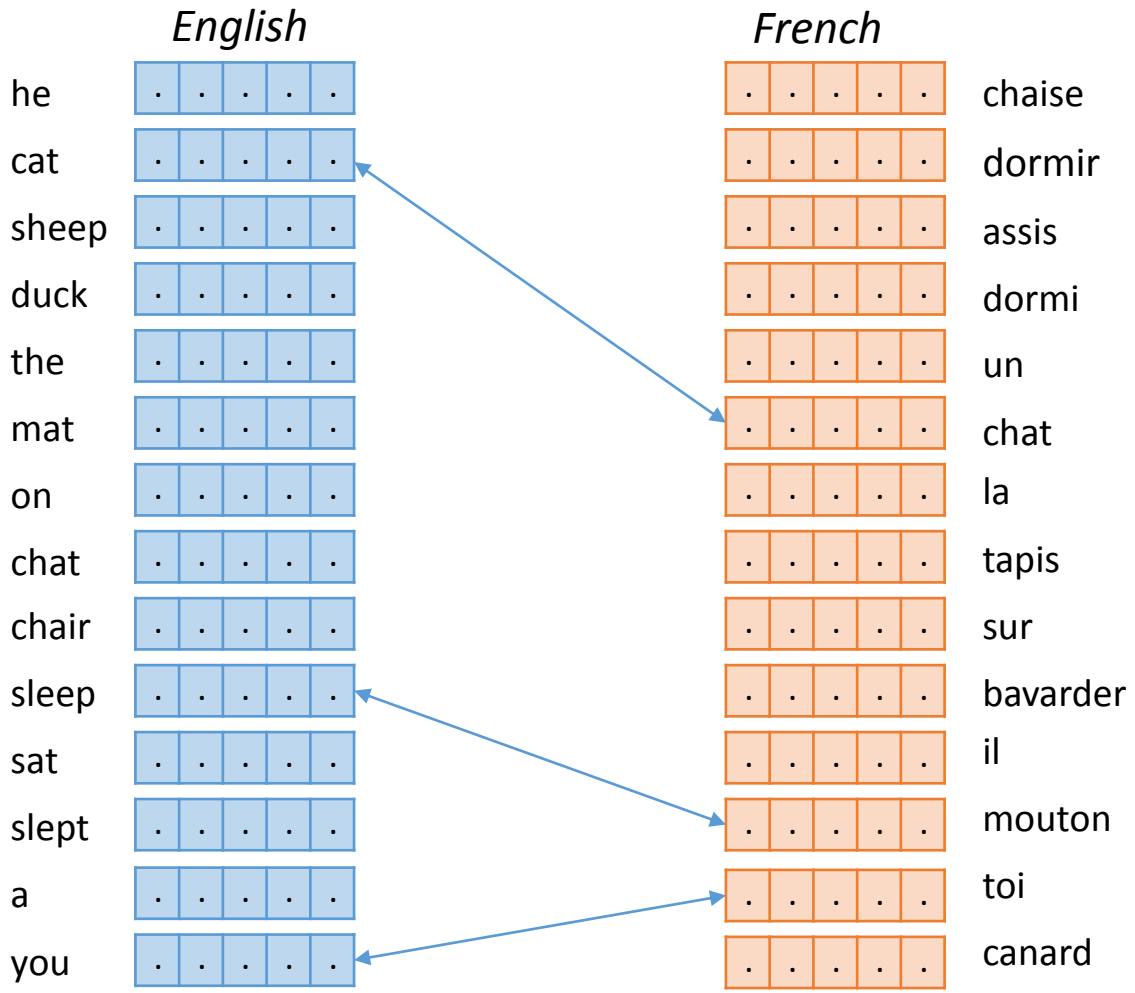
Two paradigms:

- ***Offline Bilingual Alignment***
- *Joint training for Bilingual Alignment*

***Offline bilingual alignment**:*

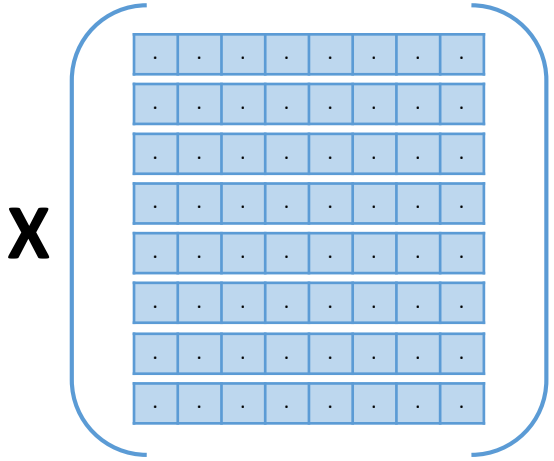
- *Stage 1: independently learn word representations for two languages*
- *Stage 2: Now try to enforce similarity between the representations of similar words across the two languages.*

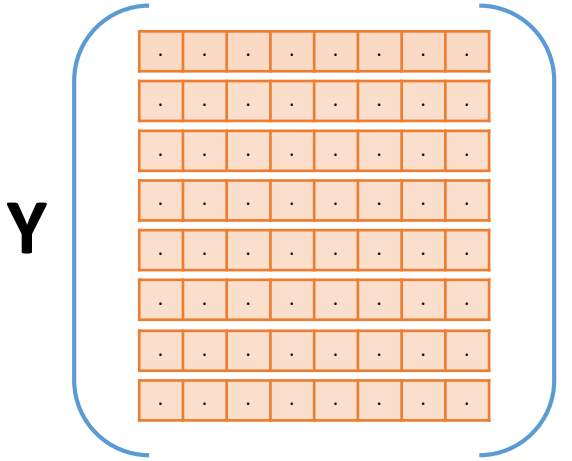
How? Let's see …



Goal in Stage 2: transform the word representations such that representations of (cat, chat), (sheep, mouton) , (you, toi), etc. are close to each other

After Stage 1: *X = representations of English words*
Y = representations of French words



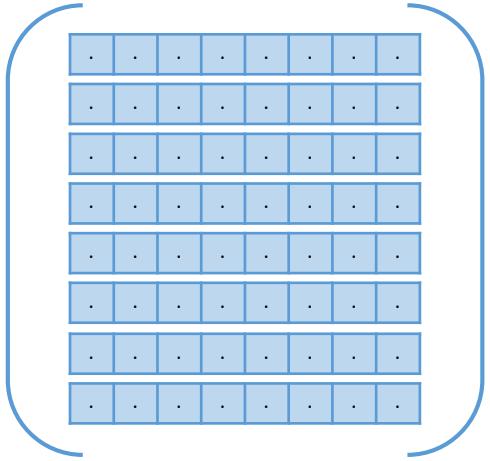


Use a bilingual dictionary to make X and Y parallel (i.e. corresponding rows in X and Y form a translation pair)

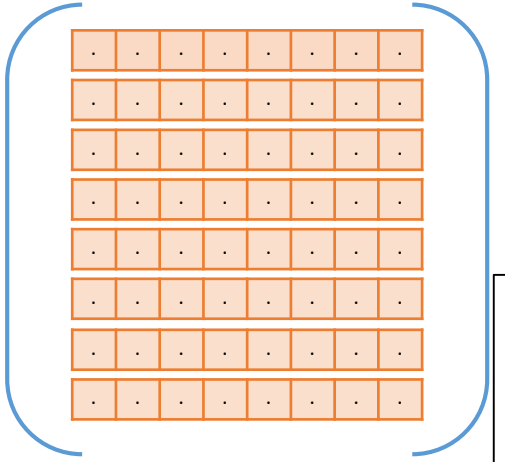
(Faruqui and Dyer, 2014)

After Stage 1: *X = representations of English words*
Y = representations of French words

X

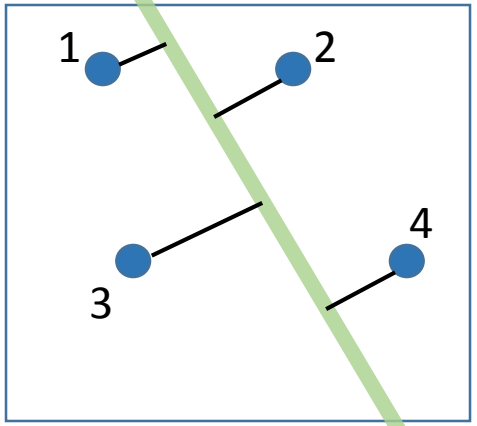


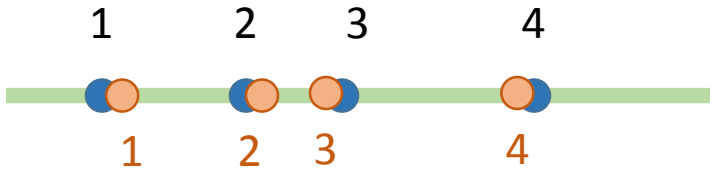
Y



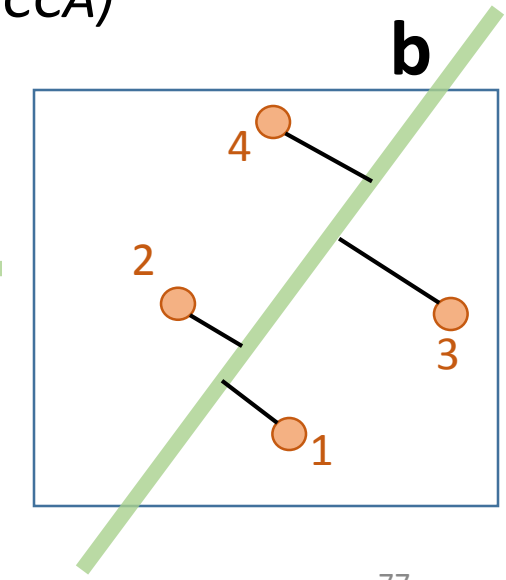
Use Canonical Correlation Analysis (CCA)

a





b



Goal *: transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*

Search for projection vectors a & b

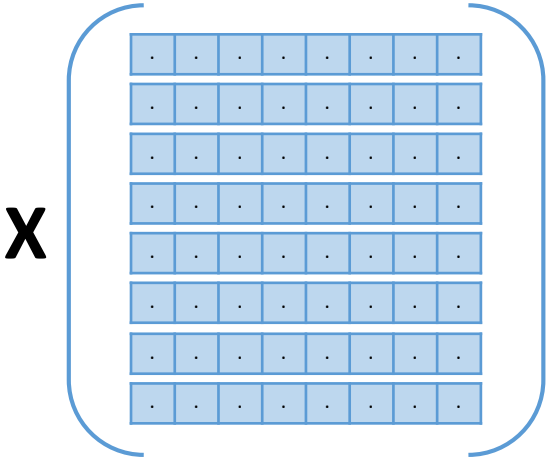
Such that after projecting the original representations on a and b …

… the projections are correlated

(Faruqui and Dyer, 2014)

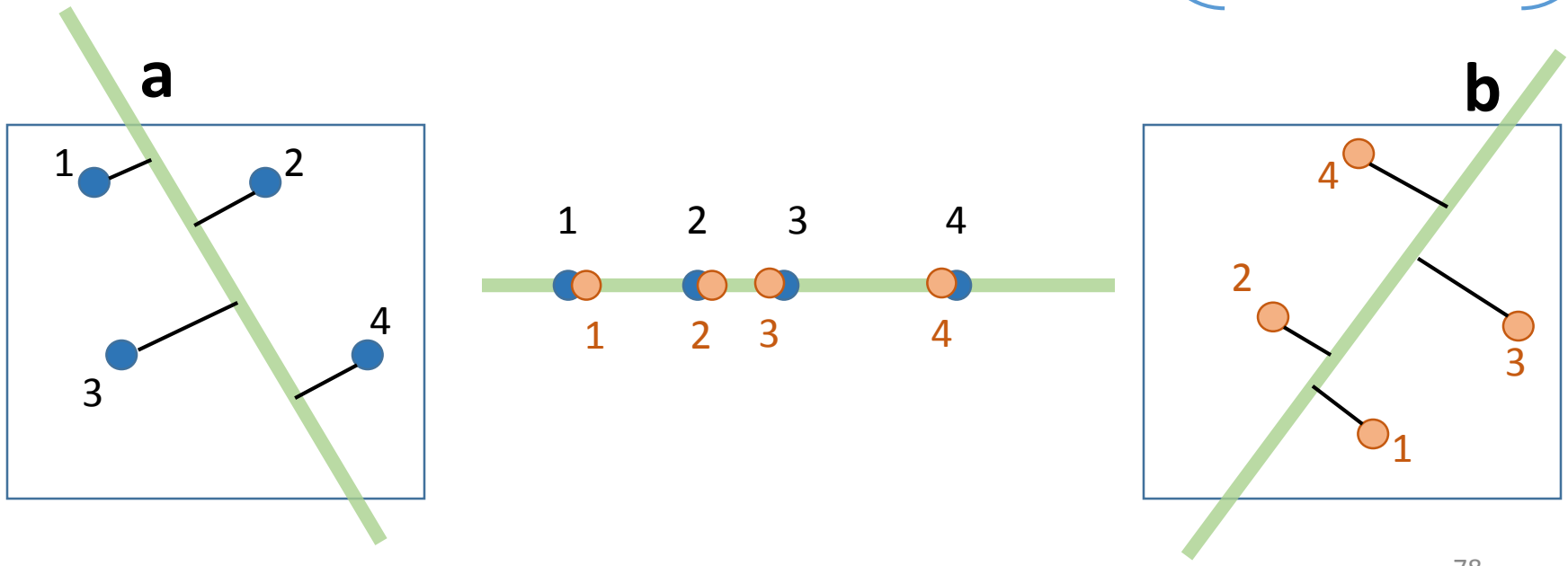
After Stage 1: *X = representations of English words*
Y = representations of French words

Goal : *transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*



Transform X by projecting it on A

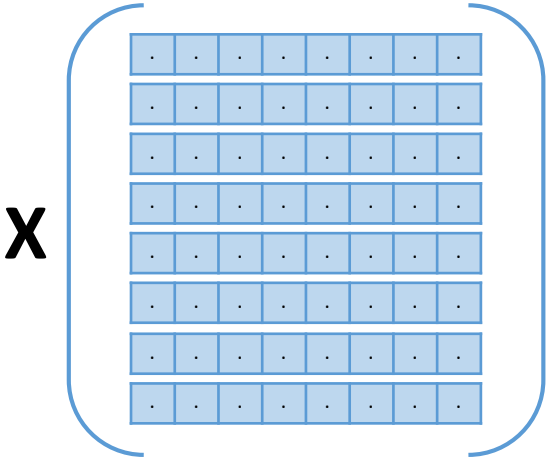
XA



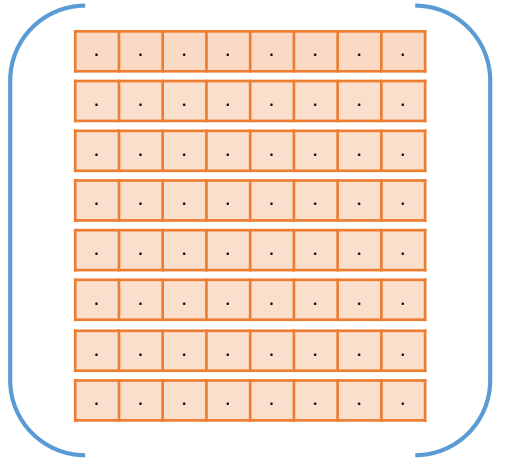
(Faruqui and Dyer, 2014)

After Stage 1: *X = representations of English words*
Y = representations of French words

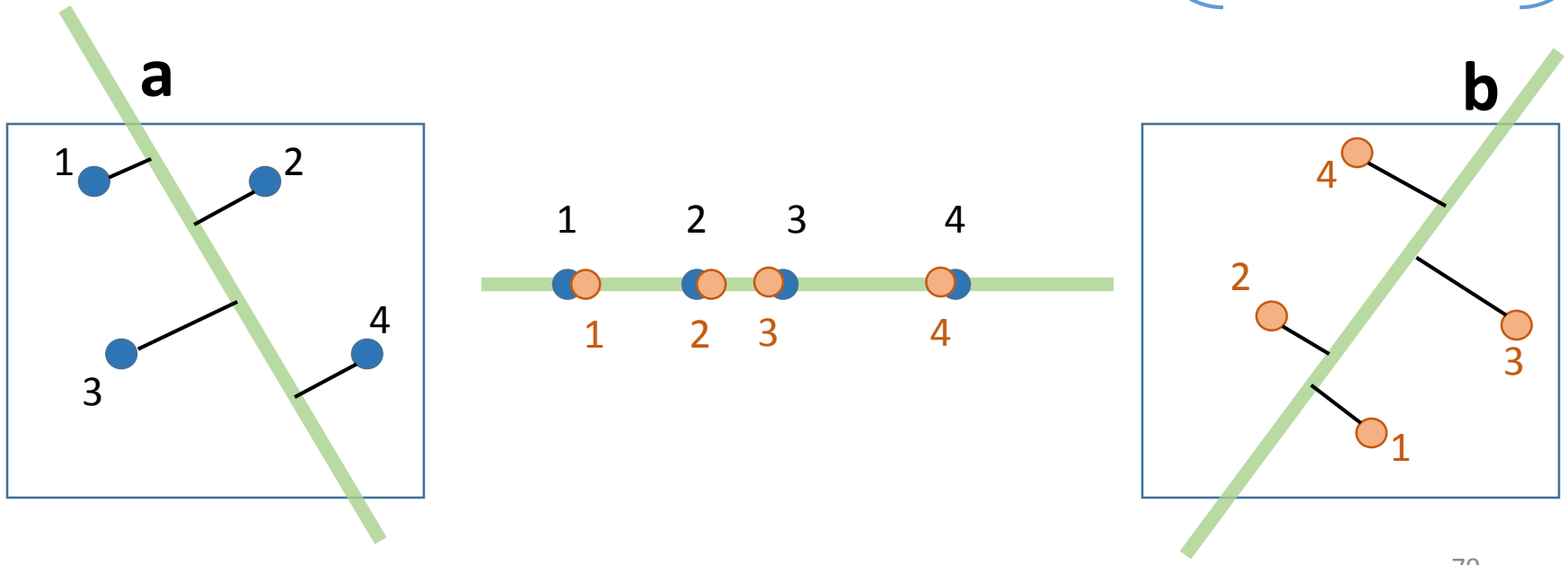
***Goal** : transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*



Transform Y by projecting it on B



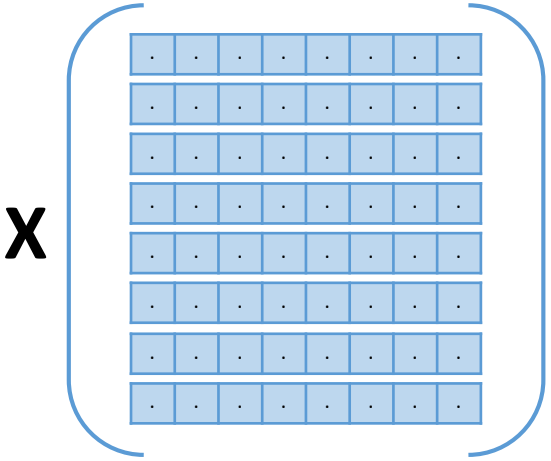
XA YB

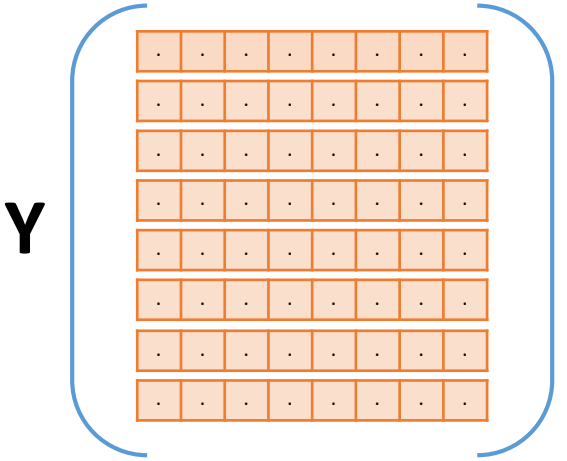


(Faruqui and Dyer, 2014)

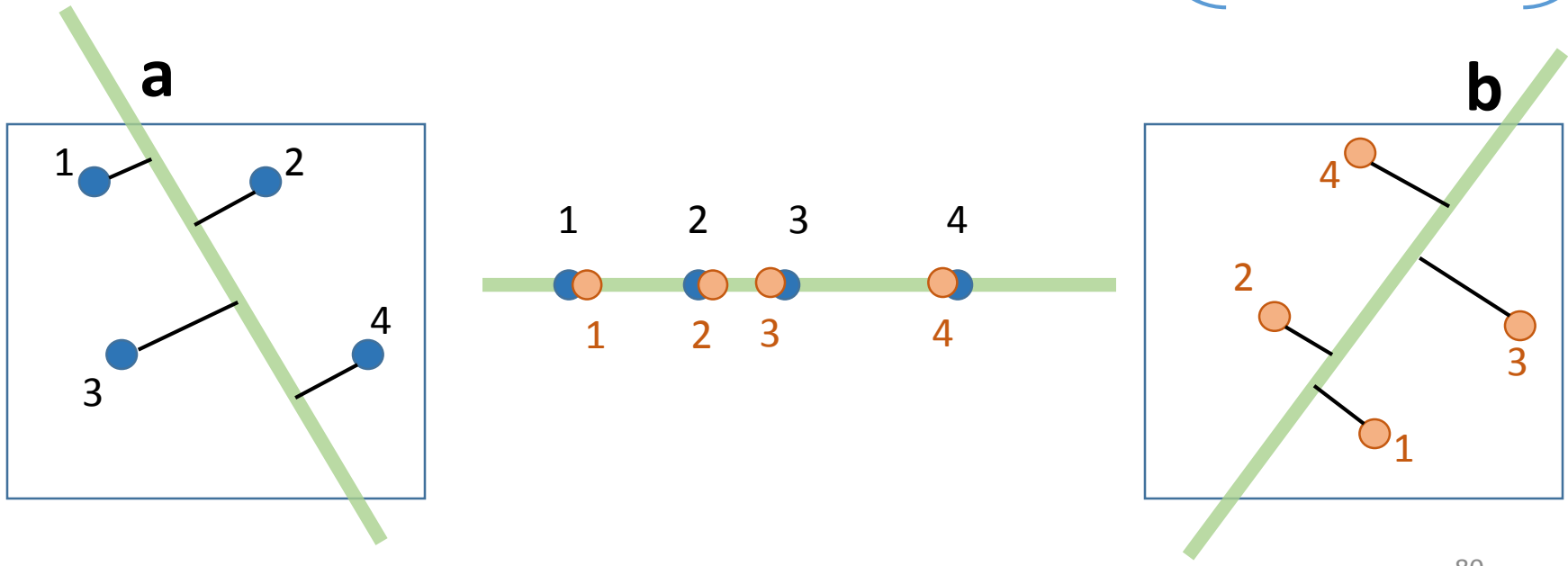
After Stage 1: *X = representations of English words*
Y = representations of French words

***Goal** : transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*





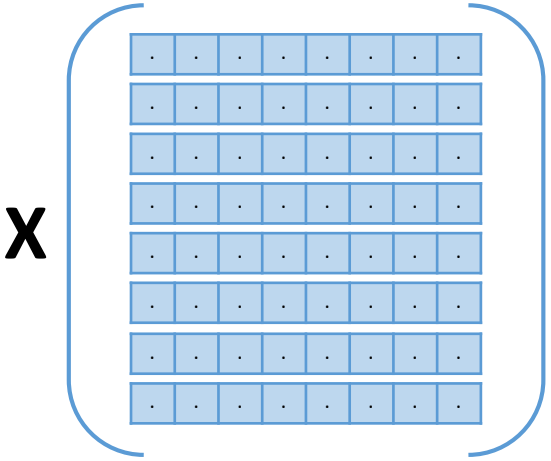
$(XA)^T\ YB$



(Faruqui and Dyer, 2014)

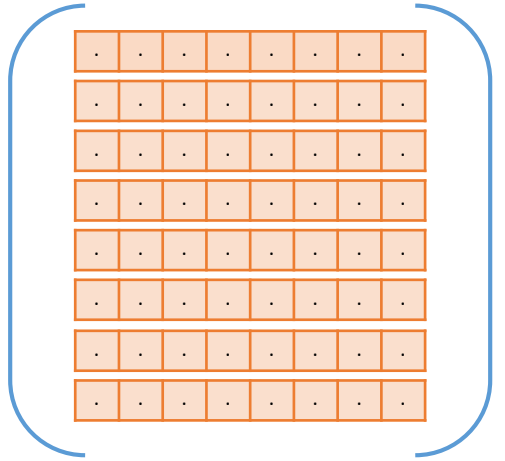
After Stage 1: *X = representations of English words*
Y = representations of French words

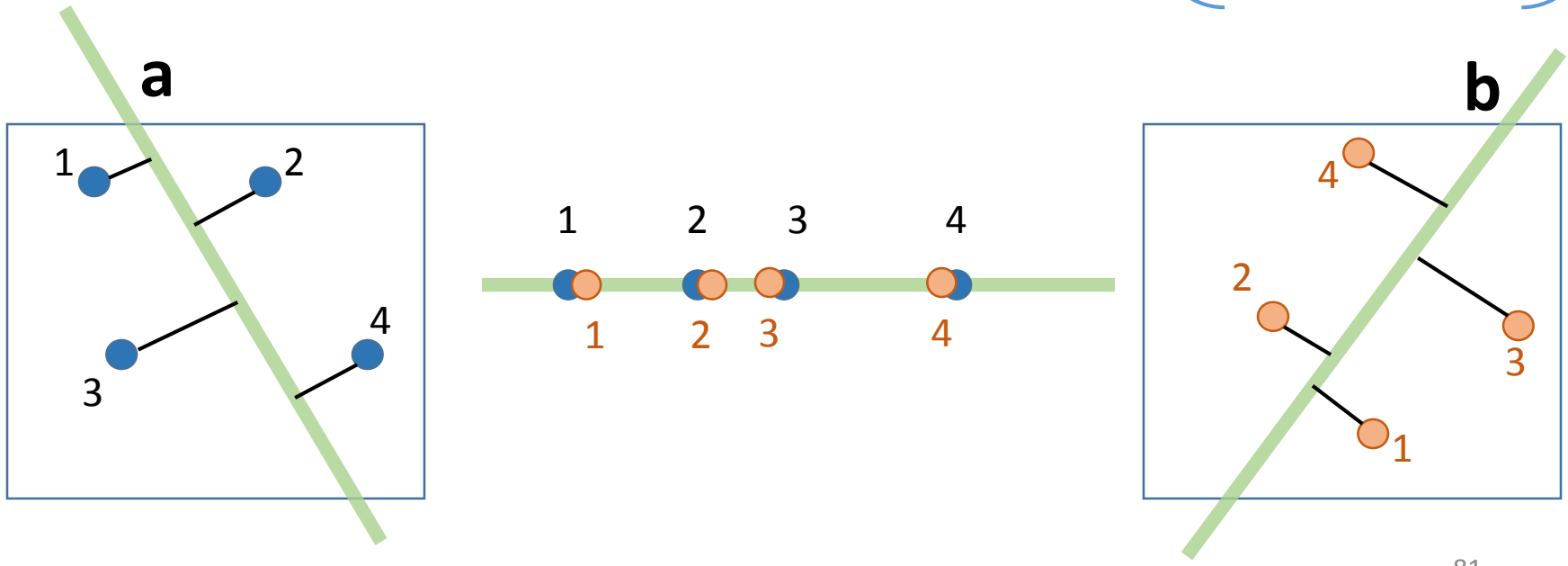
Goal : *transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*



This term is simply the correlation between transformations XA & YB. We need to maximize this

$$A^T X^T Y B$$

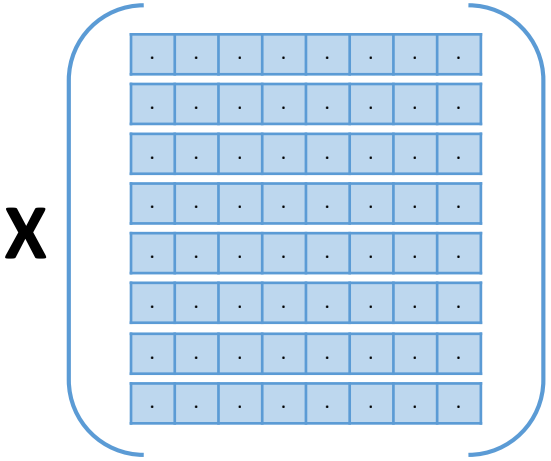


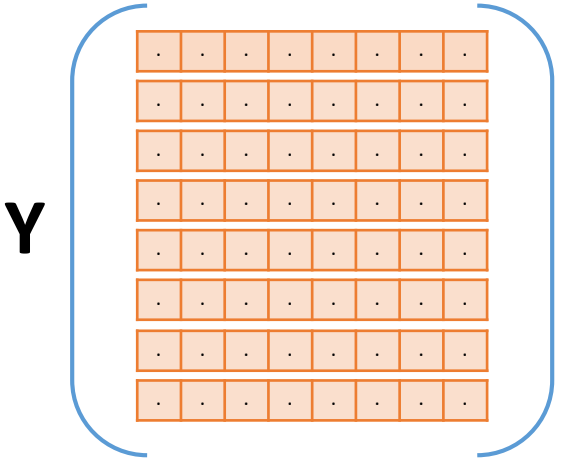


(Faruqui and Dyer, 2014)

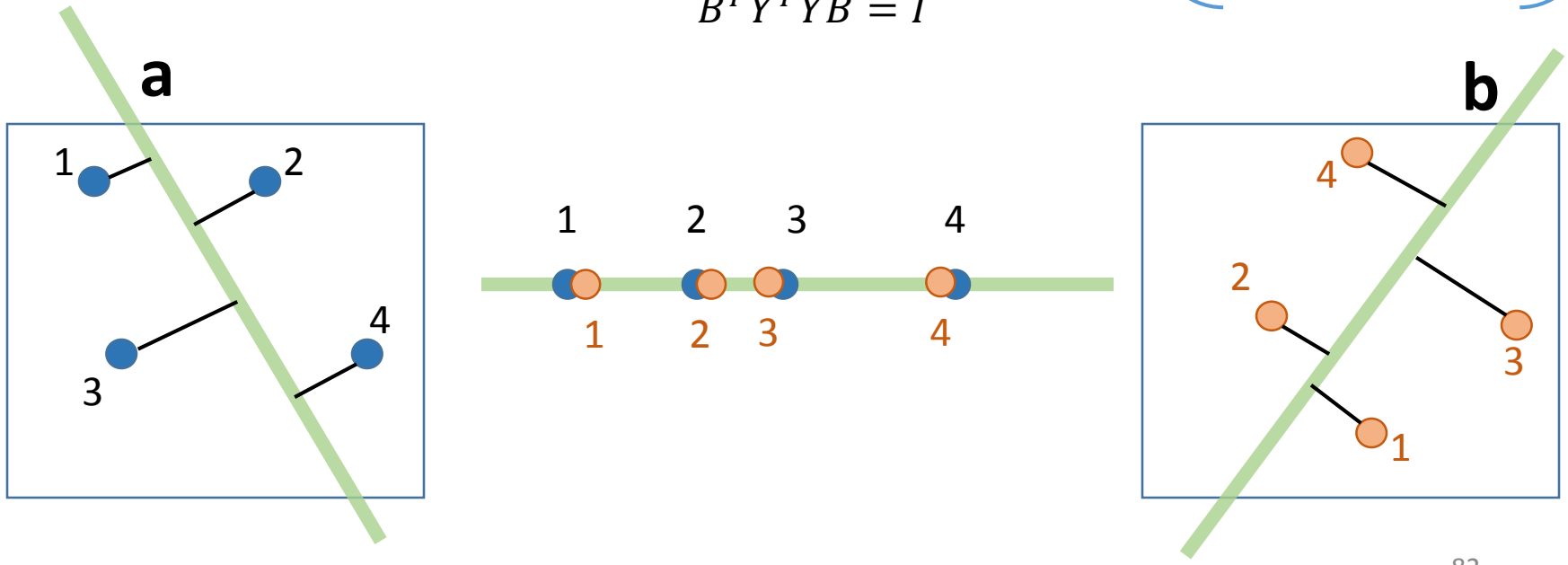
After Stage 1: *X = representations of English words*
Y = representations of French words

***Goal** : transform X and Y such that the transformed representations of (cat, chat), (you, toi), etc. are close to each other*





$$maximize\ trace(A^T X^T Y B)$$
$$s.t.\ A^T X^T X A = I$$
$$B^T Y^T Y B = I$$



(Faruqui and Dyer, 2014)

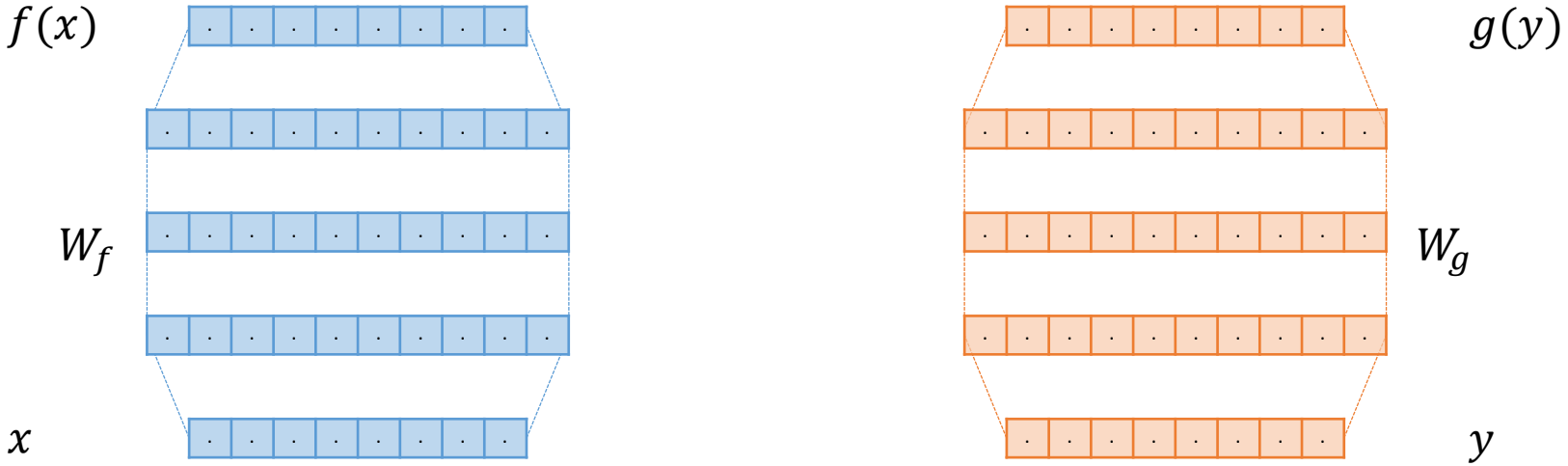
Alternately one could use Deep CCA instead of CCA …

After Stage 1: *X = representations of English words*
Y = representations of French words

$$\text{maximize} \quad \frac{a^T f(x)^T g(y) b}{\sqrt{a^T f(x)^T f(x) a}\sqrt{b^T g(y)^T g(y) b}}$$

$$w.r.t\ W_f, W_g, a, b$$

(same as CCA)



Again, use a bilingual dictionary to make X and Y parallel

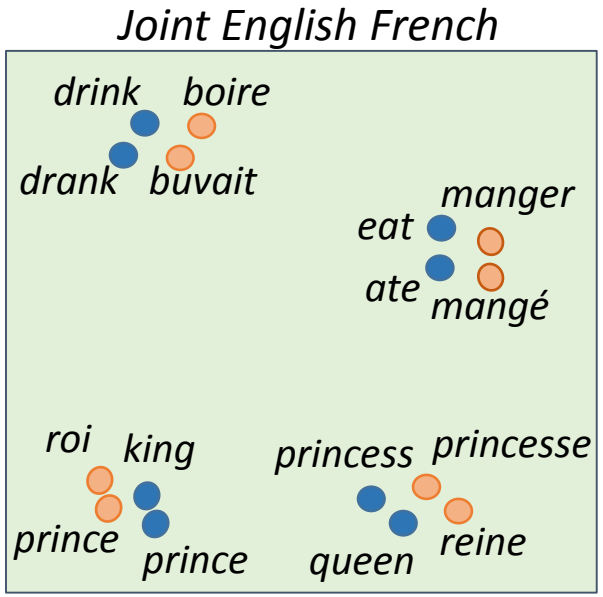
Extract deep features $f(x)$ *and* $g(y)$ *from* x *and* y

W_f *and* W_g *are the parameters of the two networks*

Now find projection vectors a *&* b *such that …..*

Backpropagate and update W_f, W_g, a, b

(Lu et. al, 2015)

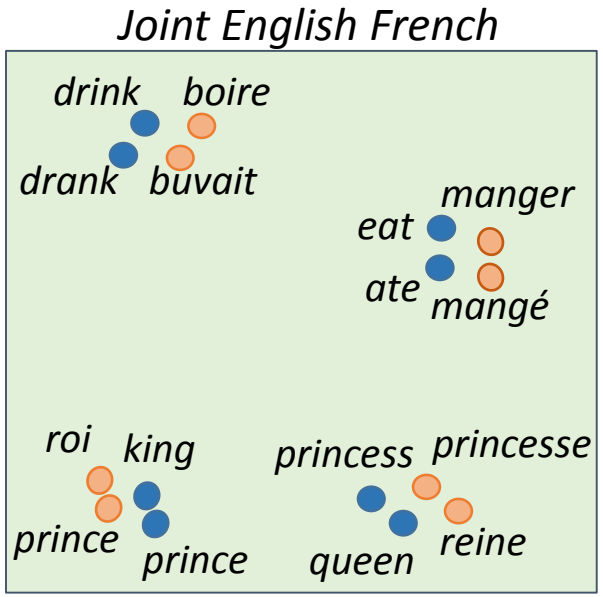


Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

Two paradigms:

- *Offline Bilingual Alignment*
- ***Joint training for Bilingual Alignment***

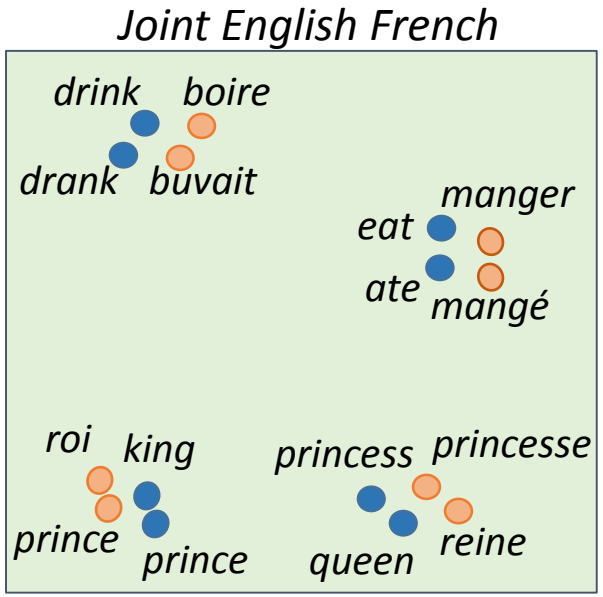


Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

Two paradigms:

- *Offline Bilingual Alignment*
- ***Joint training for Bilingual Alignment***
  - *Use only parallel data*
  - *Use monolingual as well a parallel data*



Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

Two paradigms:

- *Offline Bilingual Alignment*
- ***Joint training for Bilingual Alignment***
  - ***Use only parallel data***
  - *Use monolingual as well a parallel data*

Training data: Parallel sentences

$a = English\ sentence$
$b = parallel\ French\ sentence$
$n = random\ French\ sentence$

$minimize$
$E(a,b) = \left|\left|f(a) - g(b)\right|\right|^2$

$minimize$
$max(0, m + E(a,b) - E(a,n))$

degenerate solution is to make $f(a) = g(b) = 0$

To avoid this use max-margin training

Backpropagate & update w_i*'s in both languages*

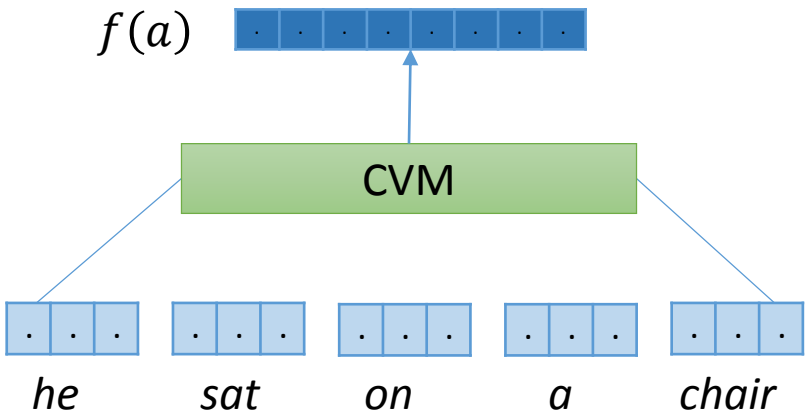
Compose word representations to get a sentence representation using a Compositional Vector Model(CVM)

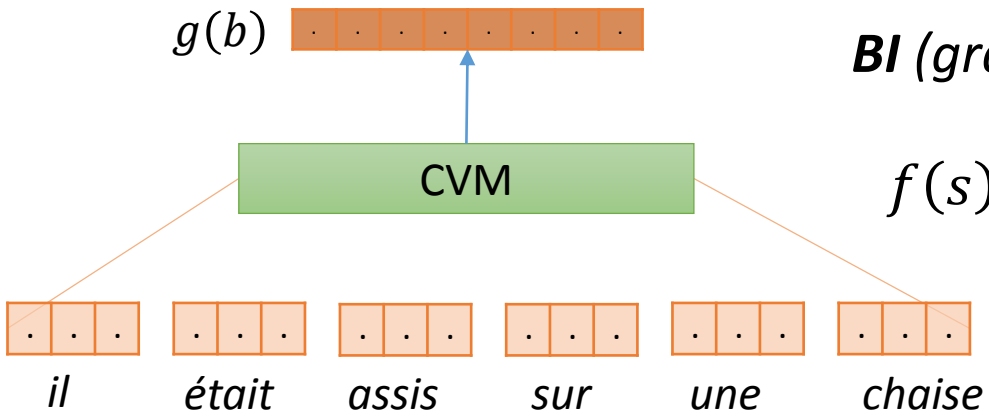
Two options considered:
***ADD**: (simply add word vectors)*
$s = sentence$
$w_i = representation\ of\ word\ i\ in\ the\ sentence$

$$f(s) = \sum_{i=1}^{n} w_i$$

***BI** (gram):*

$$f(s) = \sum_{i=1}^{n} \tanh(w_{i-1} + w_i)$$





(Hermann & Blunson, 2014)

Training data: Parallel sentences

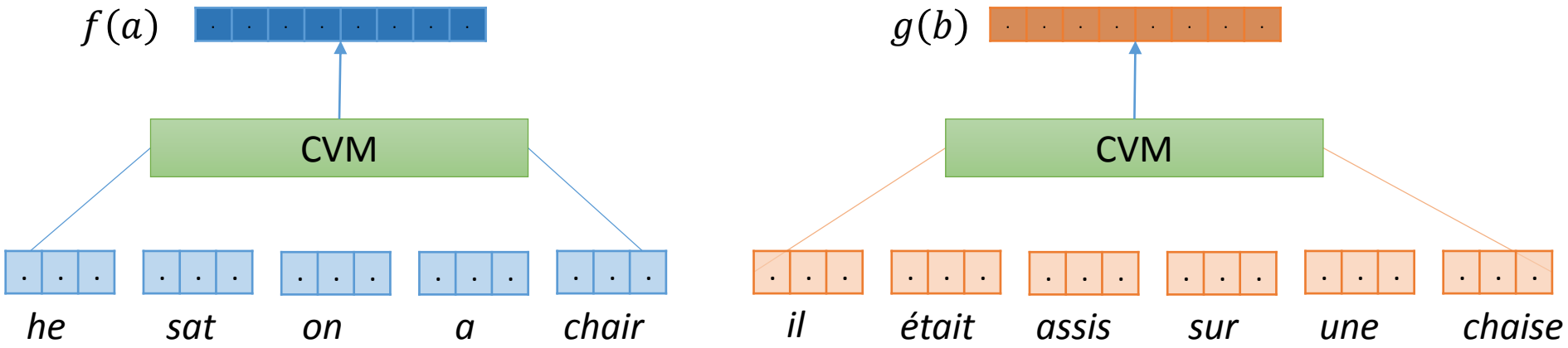
$a = English\ sentence$
$b = parallel\ French\ sentence$
$n = random\ French\ sentence$

$$E(a,b) = \left|\left|f(a) - g(b)\right|\right|^2$$

$$minimize$$
$$max(0, m + E(a,b) - E(a,n))$$

Backpropagate & update w_i's in both languages

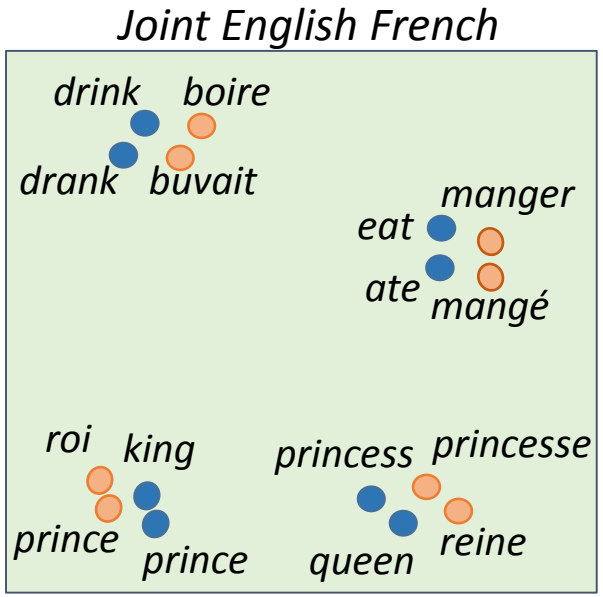
To reduce the distance between $f(a)$ & $g(b)$ the model will eventually learn to reduce the distance between (chair, chaise), (sit, assis), (he, il) etc.



(Hermann & Blunson, 2014)

The previous approach strictly requires parallel data…

Can we exploit monolingual data in two languages in addition to parallel data between them?



Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

Can also be extended to bigger units (sentences, documents, etc.)

Two paradigms:

- *Offline Bilingual Alignment*
- ***Joint training for Bilingual Alignment***
  - *Use only parallel data*
  - ***Use monolingual as well a parallel data***

Given monolingual data we already know how to learn word representations

$$\text{maximize} \sum_{i=1}^{T_e} -\log(P(w_i|w_{i-k}, \dots, w_{i-1}))$$

$$w.r.t\ W^e_{emb}, W^e_h, W^e_{out}$$

T_e = *total number of words in the English corpus*
W^e_{emb} = *word representation for English words*
W^e_h, W^e_{out} = *other parameters of the model*

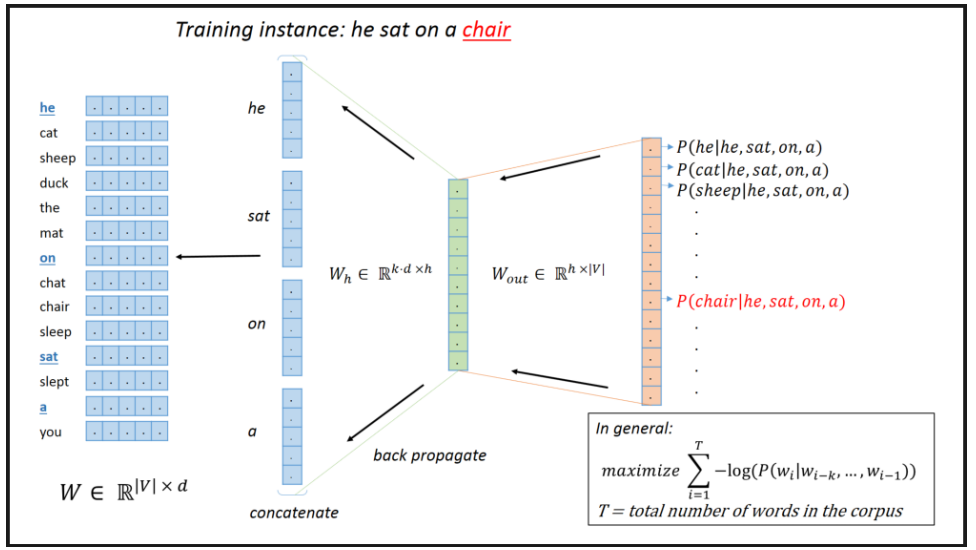
Similarly for French

$$\text{maximize} \sum_{i=1}^{T_f} -\log(P(w_i|w_{i-k}, \dots, w_{i-1}))$$

$$w.r.t W^f_{emb}, W^f_h, W^f_{out}$$

T_f = *total number of words in the French corpus*
W^f_{emb} = *word representation for French words*
W^f_h, W^f_{out} = *other parameters of the model*

Recap:



Simply putting the two languages together we get ……

$$maximize \quad \sum_{j \in \{e,f\}} \sum_{i=1}^{T_j} -\log(P(w_i | w_{i-k}, \ldots, w_{i-1}))$$

$$w.r.t \;\; \theta_e, \theta_f$$

$$\theta_e = W^e_{emb}, W^e_h, W^e_{out}$$

$$\theta_f = W^f_{emb} W^f_h, W^f_{out}$$

Nothing great about this… this is same as training θ_e, θ_f separately

Things become interesting when in addition we have parallel data…

We can then modify the objective function ….

$$maximize \sum_{j\in\{e,f\}} \sum_{i=1}^{T_j} \underbrace{-\log(P(w_i|w_{i-k},\dots,w_{i-1}))}_{\text{monolingual similarity}} + \underbrace{\lambda \cdot \Omega\,(W^e_{emb}, W^f_{emb})}_{\text{bilingual similarity}}$$

$w.r.t\ \ \theta_e, \theta_f$

$\theta_e = W^e_{emb}, W^e_h, W^e_{out}$

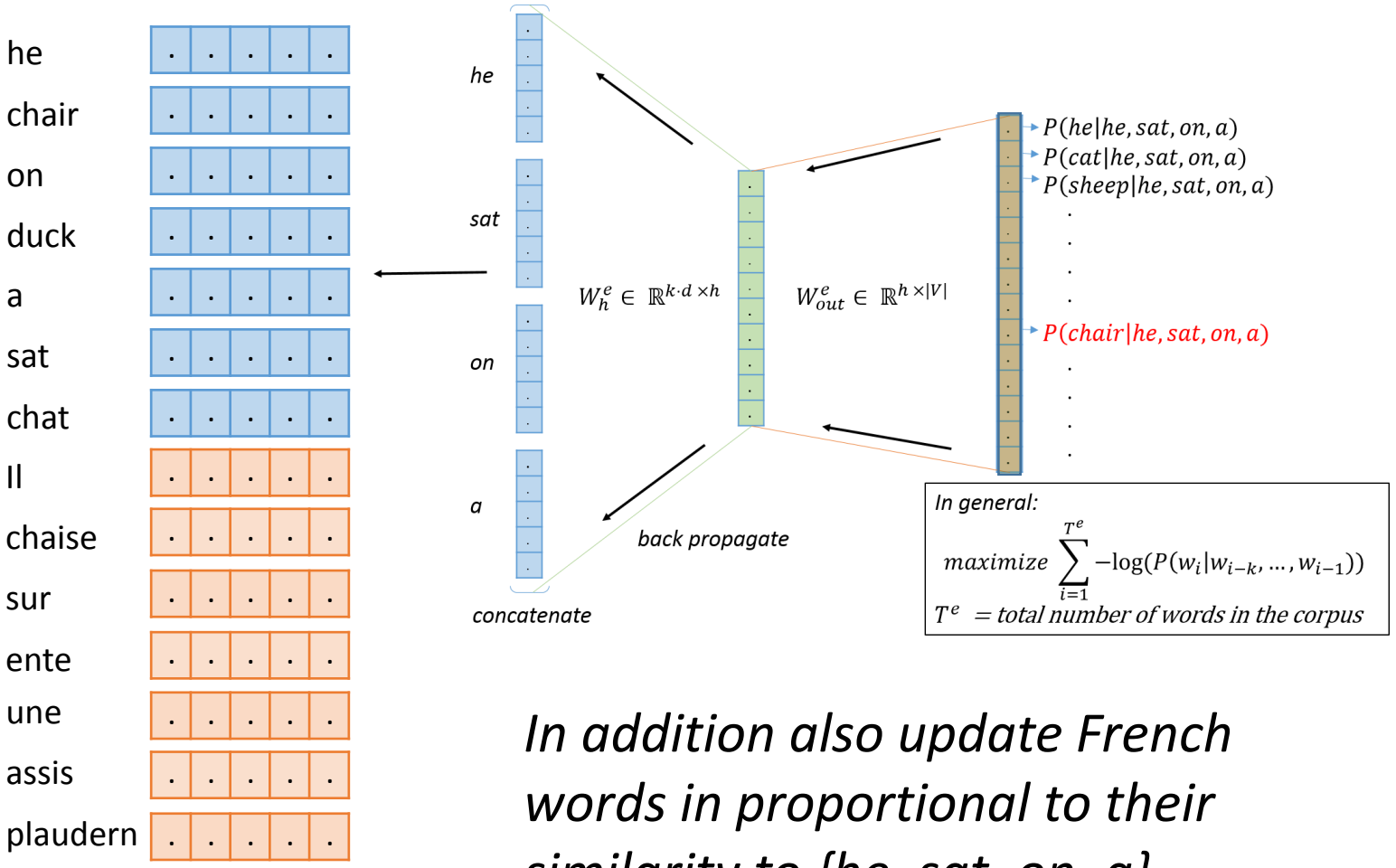
$\theta_f = W^f_{emb}, W^f_h, W^f_{out}$

$$\Omega\left(W^e_{emb}, W^f_{emb}\right) = \sum_{w_i\in V^e} \sum_{w_j\in V^f} sim(w_i, w_j) * distance(W^e_{emb_i}, W^f_{emb_j})$$

This weighted sum will be low only when similar words across languages are embedded close to each other

Now, lets look at two specific instances of this formulation….

English Training instance: he sat on a chair



$$W_{emb}^e \in \mathbb{R}^{|V^e| \times d}$$

$$W_{emb}^f \in \mathbb{R}^{|V^f| \times d}$$

In addition also update French words in proportional to their similarity to {he, sat, on, a}

(Klementiev et. al., 2012)

	assis	il	une	sur	chaise
he	0.02	0.9	0.05	0.01	0.02
sat	0.85	0.01	0.02	0.03	0.09
chair	0.06	0.01	0.01	0.01	0.95
a	0.02	0.02	0.92	0.02	0.02
on	0.10	0.05	0.05	0.81	0.04

A

$Each\ cell\ (i,j)\ of\ A\ stores\ sim(w_i, w_j)$ using word alignment information from a parallel corpus

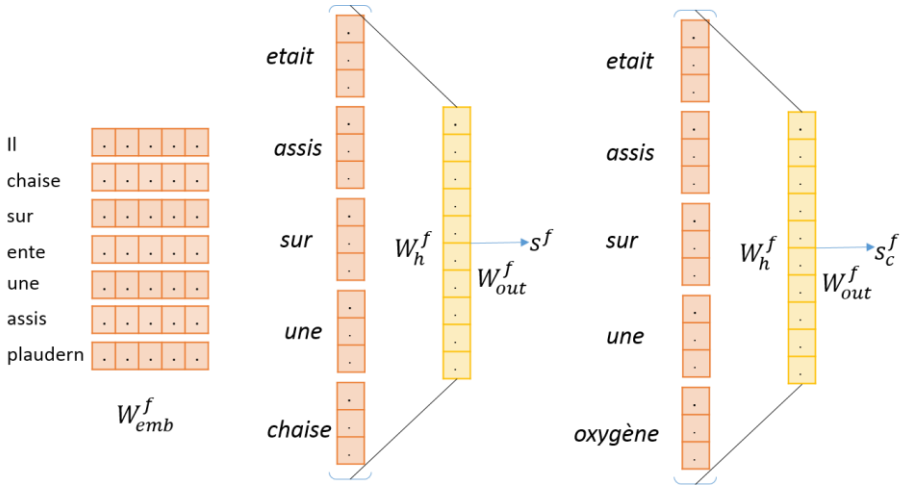
$More\ formally,$

$$W_{emb_i}^f = W_{emb_i}^f + \sum_{w_j \in V^e} A_{i,j} \frac{\partial\, \mathcal{L}(\theta^e)}{\partial\, W_{emb_j}^e}$$

$$\mathcal{L}(\theta^e) = \sum_{i=1}^{T_e} -\log(P(w_i|w_{i-k}, \dots, w_{i-1}))$$

Similar words across the two languages undergo similar updates and hence remain close to each other

Fr positive: Il était assis sur une chaise
Fr negative: Il était assis sur une oxygène

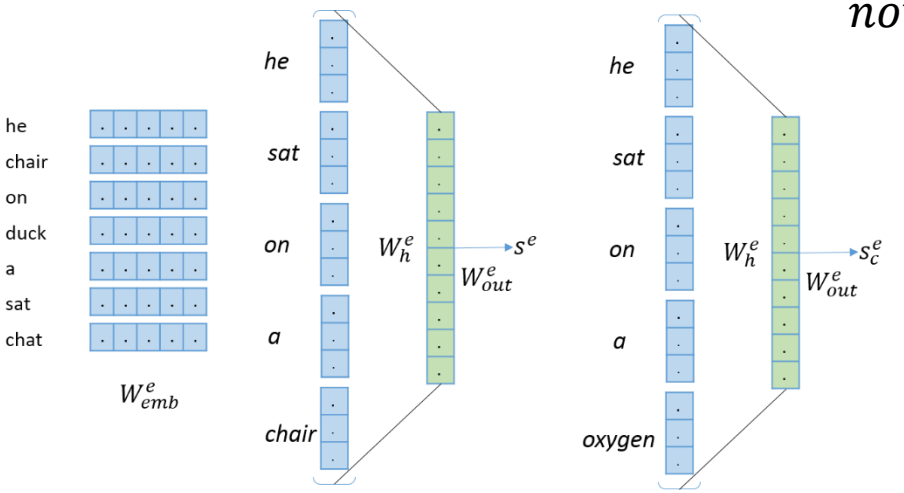


Independently update θ^e and θ^f

$$maximize\ max(0, 1 - s^f + s^f_c)$$
$$w.r.t.\ \theta^e$$

+ Parallel data
En: he sat on a chair $[s_e = w^e_1, w^e_2, w^e_3, w^e_4, w^e_5]$
Fr : Il était assis sur une chaise $[s_f = w^f_1, w^f_2, w^f_3, w^f_4, w^f_5]$

En positive: he sat on a chair
En negative: he sat on a oxygen



$$now, also\ minimize \quad \Omega\left(W^e_{emb}, W^f_{emb}\right) = \left\| \frac{1}{m} \sum_{w_i \in s^e}^{w_m} W^e_{emb_i} - \frac{1}{n} \sum_{w_j \in s^e}^{w_n} W^f_{emb_i} \right\|^2$$

$$w.r.t\ W^e_{emb}, W^f_{emb}$$

$$maximize\ max(0, 1 - s^e + s^e_c)$$
$$w.r.t.\ \theta^f$$

(Gouws et. al., 2015)

In fact, looking back, we can analyze all the approaches under this framework…

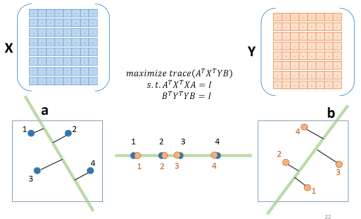
$$maximize \sum_{j \in \{e,f\}} \sum_{i=1}^{T_j} \underbrace{\mathcal{L}(\theta^j)}_{\text{monolingual similarity}} + \underbrace{\lambda \cdot \Omega\,(W^e_{emb}, W^f_{emb})}_{\text{bilingual similarity}}$$

$$w.r.t\ \ \theta_e, \theta_f$$

$$\theta_e = W_e, W_h^e, W_{out}^e$$

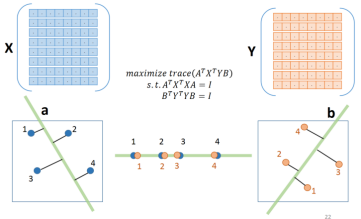
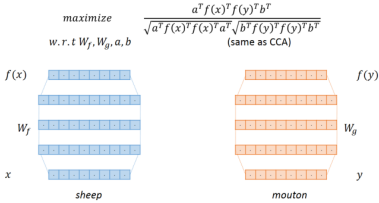
$$\theta_f = W_f, W_h^f, W_{out}^f$$

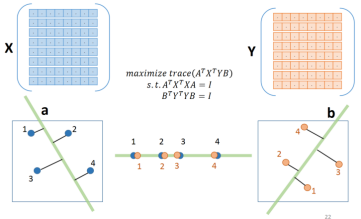
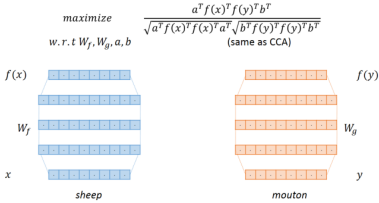
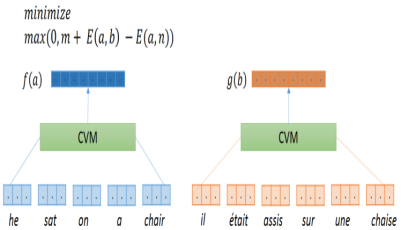
$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega\left(\theta^e, \theta^f\right)$	Training

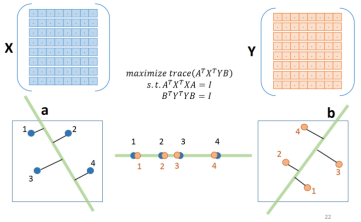
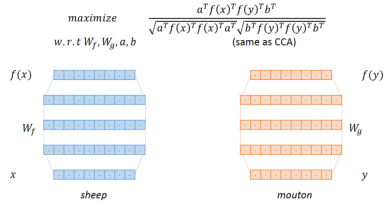
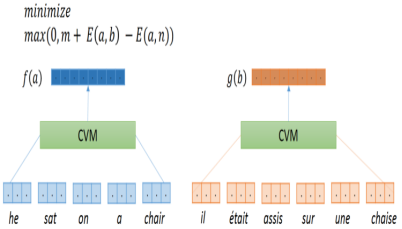
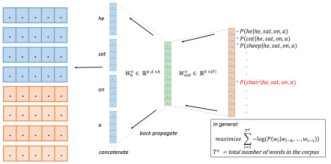


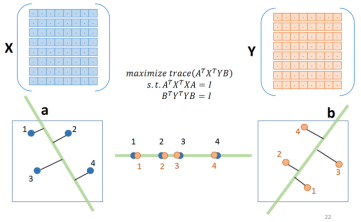
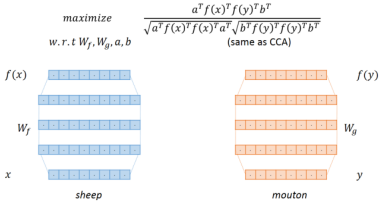
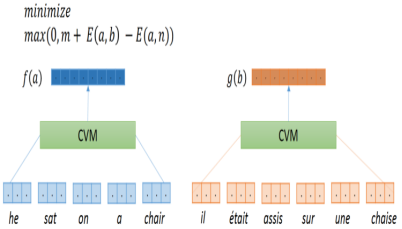
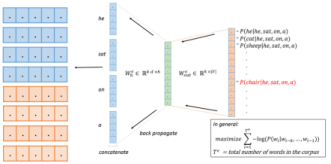
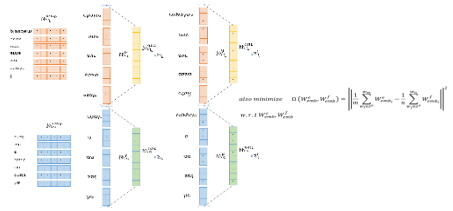
(Faruqui and Dyer, 2014)

$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega(\theta^e, \theta^f)$	Training
$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \ldots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \ldots, w_{i-1}))$	$\frac{a^T (W_{emb}^e)^T \left(W_{emb}^f\right) b}{\sqrt{a^T (W_{emb}^e)^T (W_{emb}^e) a} \sqrt{b^T \left(W_{emb}^f\right)^T \left(W_{emb}^f\right) b}}$	$\Omega(\theta^e, \theta^f)$ is optimized after optimizing $\mathcal{L}(\theta^i)$

	$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega\left(\theta^e, \theta^f\right)$	Training
 *(Faruqui and Dyer, 2014)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T (W_{emb}^e)^T \left(W_{emb}^f\right) b}{\sqrt{a^T (W_{emb}^e)^T (W_{emb}^e) a} \sqrt{b^T \left(W_{emb}^f\right)^T \left(W_{emb}^f\right) b}}$	$\Omega\left(\theta^e, \theta^f\right)$ is optimized after optimizing $\mathcal{L}(\theta^{\mathrm{i}})$
 *(Lu et. al., 2015)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T f(W_{emb}^e)^T \mathrm{g}\left(W_{emb}^f\right) b}{\sqrt{a^T f(W_{emb}^e)^T f((W_{emb}^e) a} \sqrt{b^T f\left(W_{emb}^f\right)^T f\left(W_{emb}^f\right) b}}$	-- " --

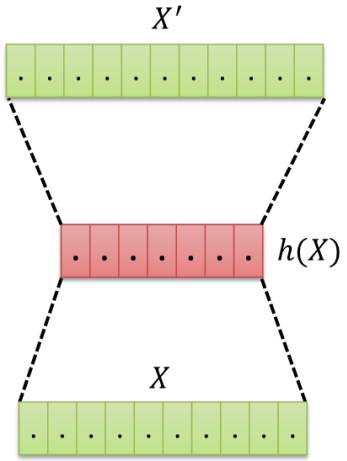
	$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega\left(\theta^e, \theta^f\right)$	Training
 *(Faruqui and Dyer, 2014)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T (W^e_{emb})^T \left(W^f_{emb}\right) b}{\sqrt{a^T (W^e_{emb})^T (W^e_{emb}) a} \sqrt{b^T \left(W^f_{emb}\right)^T \left(W^f_{emb}\right) b}}$	$\Omega\left(\theta^e, \theta^f\right)$ is optimized after optimizing $\mathcal{L}\left(\theta^{\mathrm{i}}\right)$
 *(Lu et. al., 2015)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T f(W^e_{emb})^T \mathrm{g}\left(W^f_{emb}\right) b}{\sqrt{a^T f(W^e_{emb})^T f((W^e_{emb}) a} \sqrt{b^T f\left(W^f_{emb}\right)^T f\left(W^f_{emb}\right) b}}$	-- " --
 *(Herman & Blunsom, 2014)*	0	0	$max(0, m + E(a,b) - E(a,n))$ $where, E(a,b) = \left\|\left\|f(a) - g(b)\right\|\right\|^2$	Only $\Omega\left(\theta^e, \theta^f\right)$ is optimized

	$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega\left(\theta^e, \theta^f\right)$	Training
 *(Faruqui and Dyer, 2014)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T (W^e_{emb})^T \left(W^f_{emb}\right) b}{\sqrt{a^T (W^e_{emb})^T (W^e_{emb}) a} \sqrt{b^T \left(W^f_{emb}\right)^T \left(W^f_{emb}\right) b}}$	$\Omega\left(\theta^e, \theta^f\right)$ is optimized after optimizing $\mathcal{L}(\theta^{\mathrm{i}})$
 *(Lu et. al., 2015)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T f(W^e_{emb})^T \mathrm{g}\left(W^f_{emb}\right) b}{\sqrt{a^T f(W^e_{emb})^T f((W^e_{emb}) a} \sqrt{b^T f\left(W^f_{emb}\right)^T f\left(W^f_{emb}\right) b}}$	-- " --
 *(Herman & Blunsom, 2014)*	0	0	$max(0, m + E(a,b) - E(a,n))$ $where, E(a,b) = \left\|\|f(a) - g(b)\|\right\|^2$	Only $\Omega\left(\theta^e, \theta^f\right)$ is optimized
 *(Klementiev et. al., 2012)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$0.5 * W^{eT}_{emb} (A \otimes I) W^f_{emb}$	$\Omega\left(\theta^e, \theta^f\right)$ & $\mathcal{L}(\theta^{\mathrm{i}})$ are optimized jointly

	$\mathcal{L}(\theta^e)$	$\mathcal{L}(\theta^f)$	$\Omega\left(\theta^e, \theta^f\right)$	Training
 *(Faruqui and Dyer, 2014)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T (W^e_{emb})^T \left(W^f_{emb}\right) b}{\sqrt{a^T (W^e_{emb})^T (W^e_{emb}) a} \sqrt{b^T \left(W^f_{emb}\right)^T \left(W^f_{emb}\right) b}}$	$\Omega\left(\theta^e, \theta^f\right)$ is optimized after optimizing $\mathcal{L}(\theta^{\mathrm{i}})$
 *(Lu et. al., 2015)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\frac{a^T f(W^e_{emb})^T \mathrm{g}\left(W^f_{emb}\right) b}{\sqrt{a^T f(W^e_{emb})^T f((W^e_{emb}) a} \sqrt{b^T f\left(W^f_{emb}\right)^T f\left(W^f_{emb}\right) b}}$	-- " --
 *(Herman & Blunsom, 2014)*	0	0	$max(0, m + E(a,b) - E(a,n))$ $where, E(a,b) = \left\|\|f(a) - g(b)\|\right\|^2$	Only $\Omega\left(\theta^e, \theta^f\right)$ is optimized
 *(Klementiev et. al., 2012)*	$\sum_{i=1}^{T_e} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$\sum_{i=1}^{T_f} -\log(P(w_i \mid w_{i-k}, \dots, w_{i-1}))$	$0.5 * W^{eT}_{emb}(A \otimes I) W^f_{emb}$	$\Omega\left(\theta^e, \theta^f\right)$ & $\mathcal{L}(\theta^{\mathrm{i}})$ are optimized jointly
 *(Gouws et. al., 2015)*	$max(0, 1 - s^e + s^e_c)$	$max(0, 1 - s^f + s^f_c)$	$\left\|\left\|\frac{1}{m}\sum_{w_i \in s^e}^{w_m} W^e_{emb_i} - \frac{1}{n}\sum_{w_j \in s^e}^{w_n} W^f_{emb_i}\right\|\right\|^2$	-- " --

Now lets take a look at an approach which is based on autoencoders….

***Background**: a neural network based single view autoencoder*

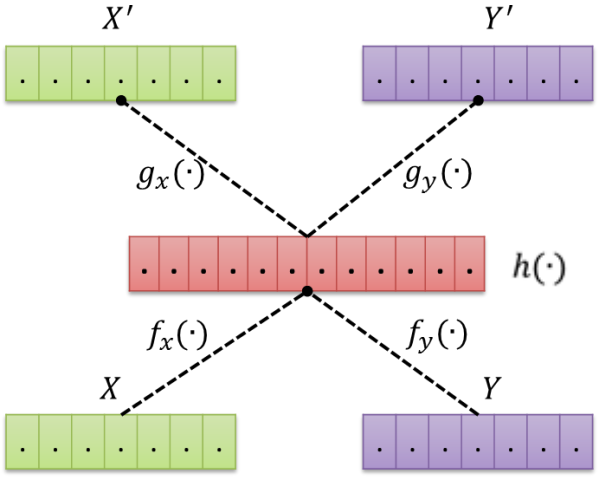


$$h(X) = f(X) = f(\boldsymbol{W}X + b) \qquad encoder$$

$$X' = \mathrm{g}(h(X)) = g(\boldsymbol{W}'h(X) + b') \qquad decoder$$

$$minimize \sum_{i=1}^{N} (X_i - g(h(X)))^2 \qquad use\ back\ propogation$$

Correlational Neural Network



A multiview autoencoder

encoder

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

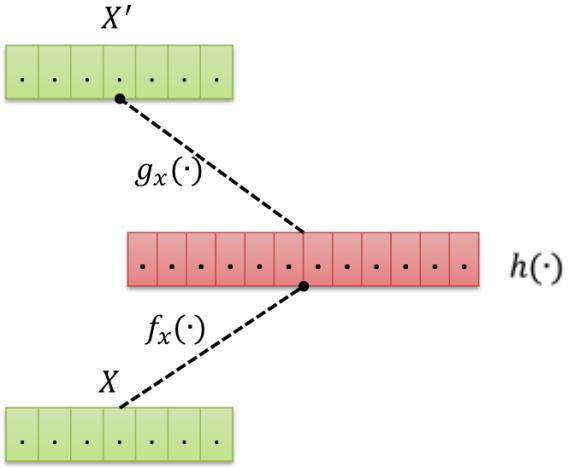
$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

decoder

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

Correlational Neural Network



A multiview autoencoder

$$minimize \sum_{i=1}^{N} (g_x(f_x(X_i)) - X_i)^2$$

encoder

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

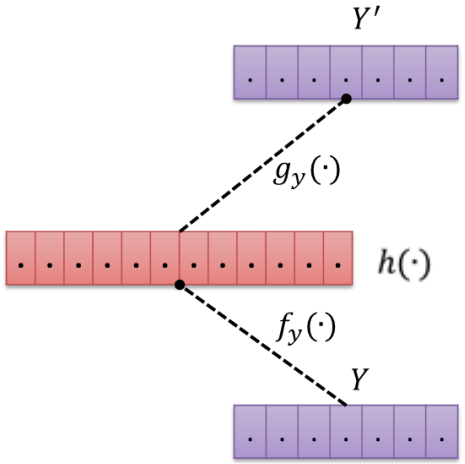
$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

decoder

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

Correlational Neural Network



A multiview autoencoder

$$\boldsymbol{encoder}$$

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

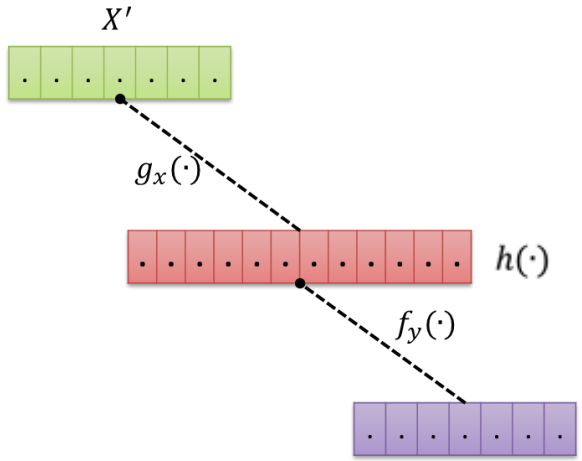
$$\boldsymbol{decoder}$$

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

$$minimize \sum_{i=1}^{N} (g_x(f_x(X_i)) - X_i)^2 + \sum_{i=1}^{N} (g_y(f_y(Y_i)) - Y_i)^2$$

Correlational Neural Network



A multiview autoencoder

$$\boldsymbol{encoder}$$

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

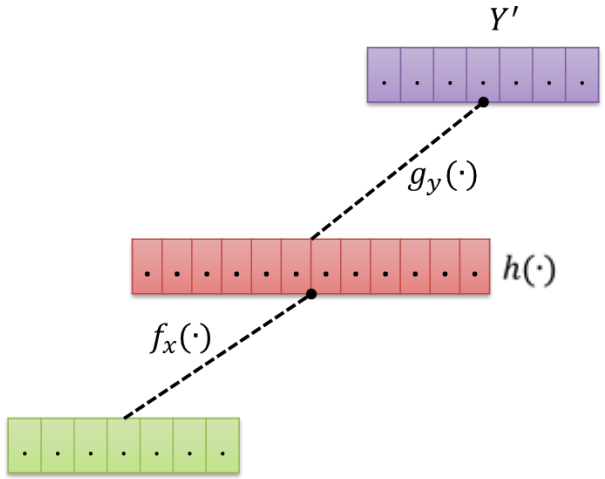
$$\boldsymbol{decoder}$$

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

$$minimize \sum_{i=1}^{N} (g_x(f_x(X_i)) - X_i)^2 + \sum_{i=1}^{N} (g_y(f_y(Y_i)) - Y_i)^2 + \sum_{i=1}^{N} (g_x(f_y(Y_i)) - X_i)^2$$

Correlational Neural Network



A multiview autoencoder

encoder

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

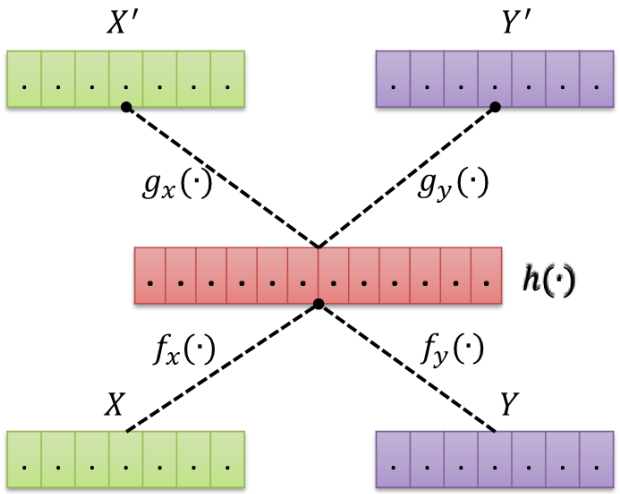
decoder

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

$$\begin{aligned} minimize \sum_{i=1}^{N} (g_x(f_x(X_i)) - X_i)^2 \\ + \sum_{i=1}^{N} (g_y(f_y(Y_i)) - Y_i)^2 \\ + \sum_{i=1}^{N} (g_x(f_y(Y_i)) - X_i)^2 \\ + \sum_{i=1}^{N} (g_y(f_x(X_i)) - Y_i)^2 \end{aligned}$$

Correlational Neural Network



A multiview autoencoder

$$\boldsymbol{encoder}$$

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

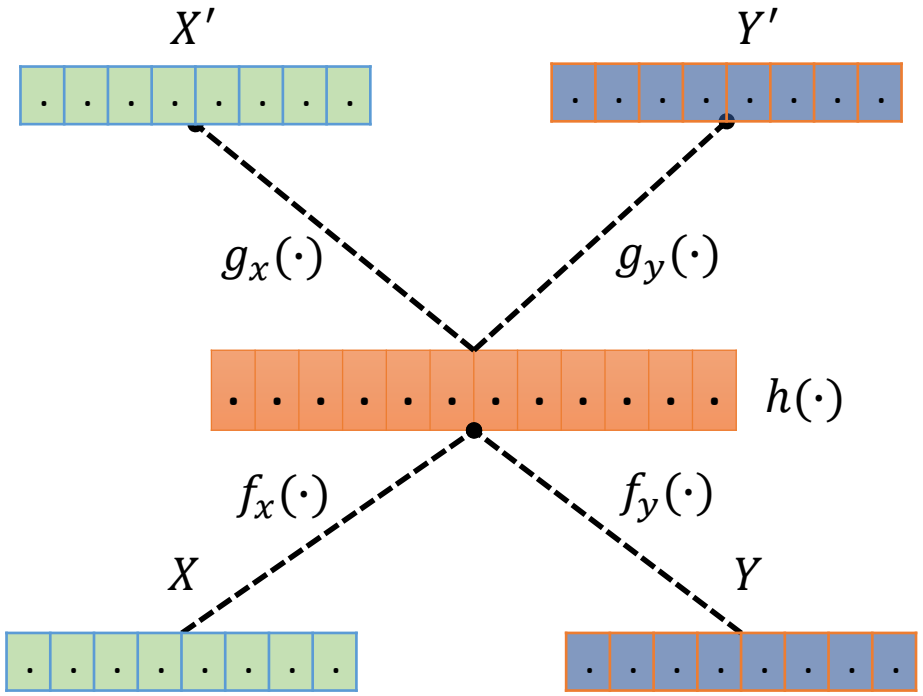
$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

$$\boldsymbol{decoder}$$

$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

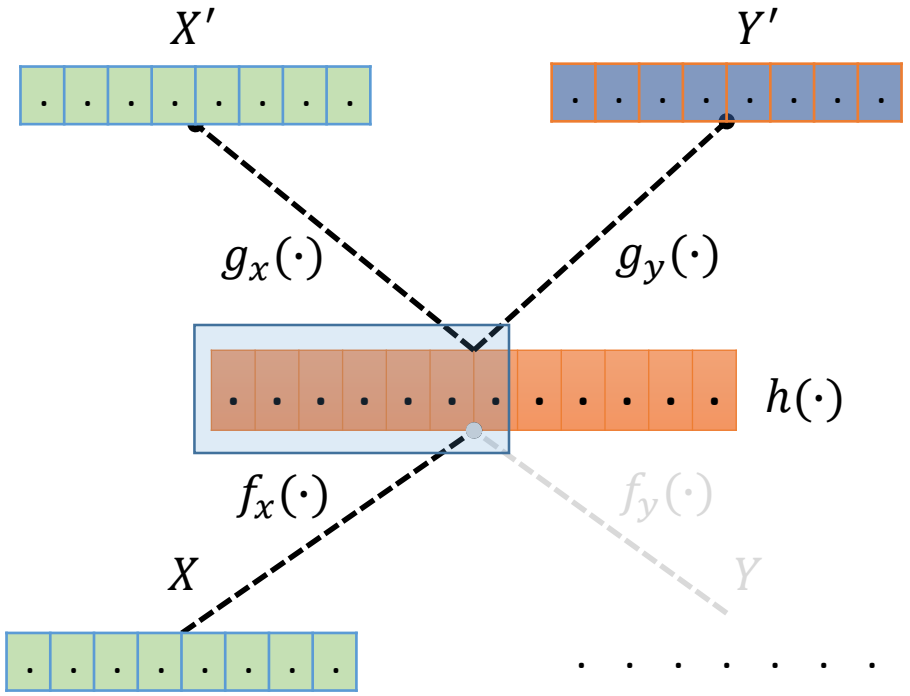
$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

$$minimize \sum_{i=1}^{N}(g_x(f_x(X_i)) - X_i)^2 + \sum_{i=1}^{N}(g_y(f_y(Y_i)) - Y_i)^2 + \sum_{i=1}^{N}(g_x(f_y(Y_i)) - X_i)^2 + \sum_{i=1}^{N}(g_y(f_x(X_i)) - Y_i)^2$$



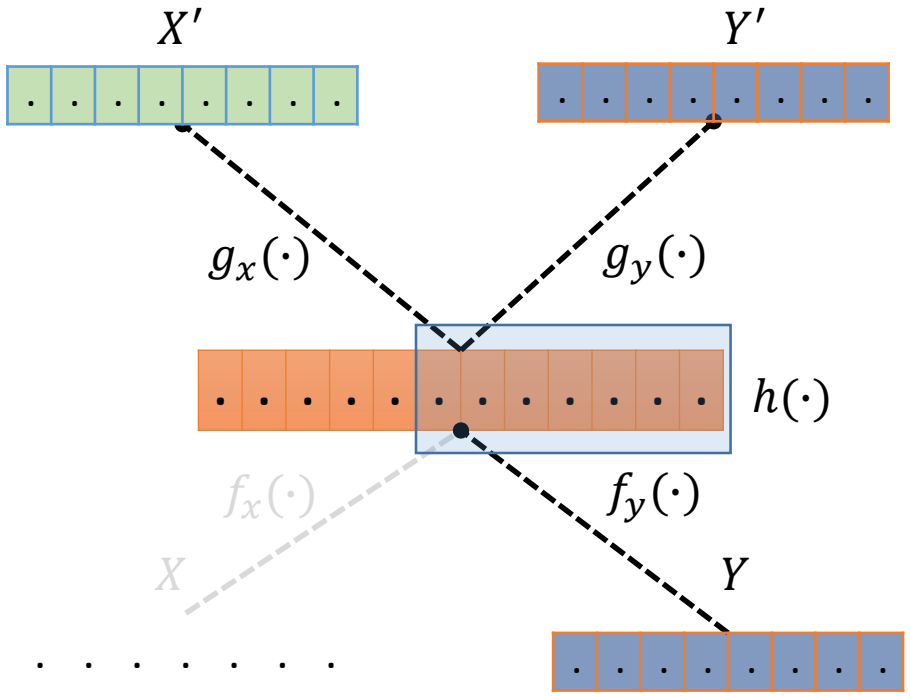
So far so good…. But will the representations h(X) and h(Y) be correlated?

Turns out that there is no guarantee for this !



So far so good…. But will the representations h(X) and h(Y) be correlated?

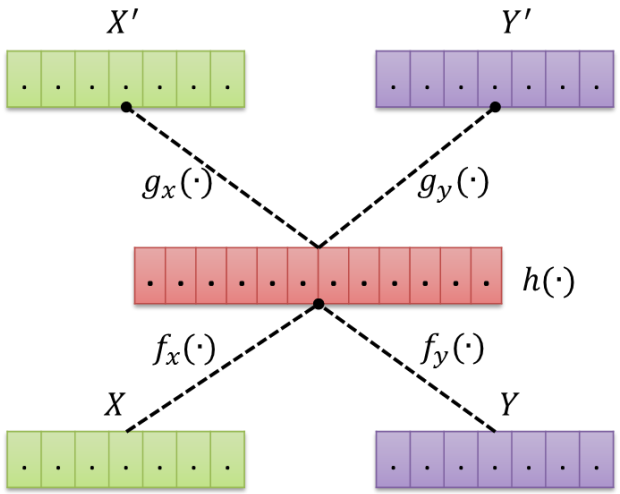
Turns out that there is no guarantee for this !



So far so good…. But will the representations h(X) and h(Y) be correlated?

Turns out that there is no guarantee for this !

Correlational Neural Network



A multiview autoencoder

$$\boldsymbol{encoder}$$

$$h_x(X) = f_x(X) = f_x(\boldsymbol{W_x}X + b)$$

$$h_y(Y) = f_y(Y) = f_y(\boldsymbol{W_y}Y + b)$$

$$\boldsymbol{decoder}$$

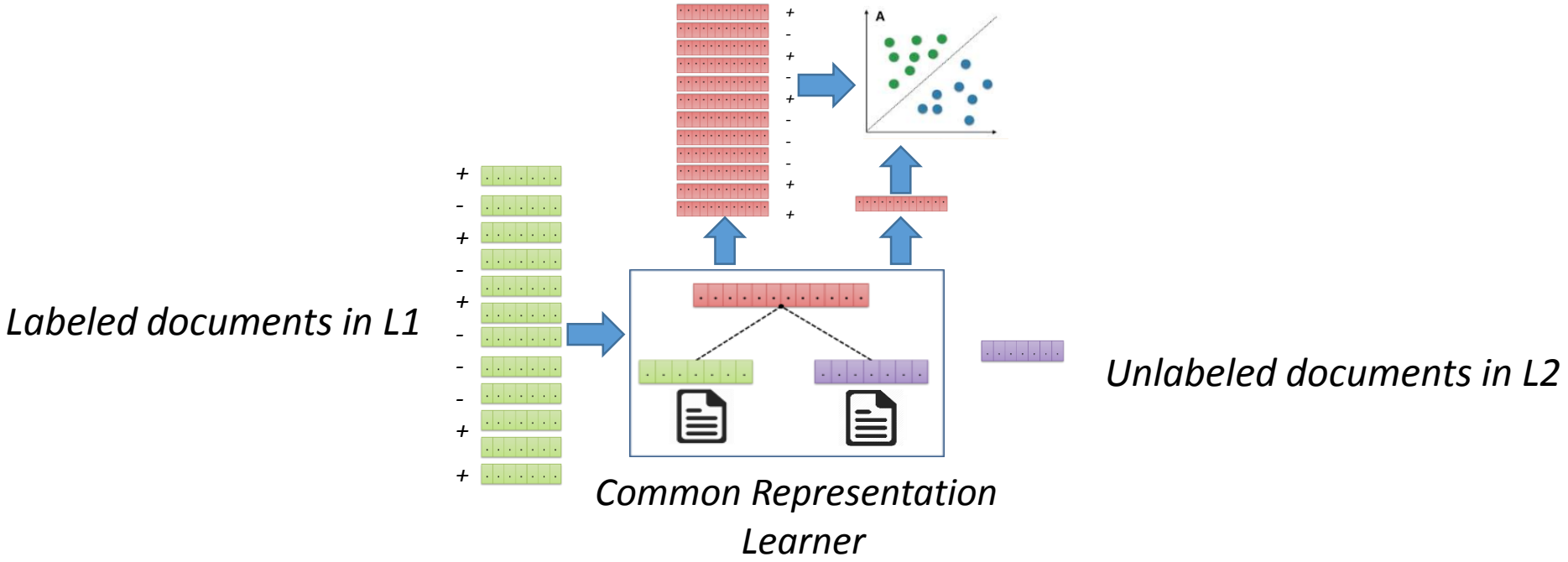
$$X' = g_x(h(X)) = g_x(\boldsymbol{W'_x}h_x(X) + b')$$

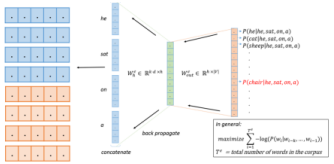
$$Y' = g_y(h(Y)) = g_y(\boldsymbol{W'_y}h_y(Y) + b')$$

$$minimize \sum_{i=1}^{N}(g_x(f_x(X_i)) - X_i)^2 + \sum_{i=1}^{N}(g_y(f_y(Y_i)) - Y_i)^2 + \sum_{i=1}^{N}(g_x(f_y(Y_i)) - X_i)^2 + \sum_{i=1}^{N}(g_y(f_x(X_i)) - Y_i)^2$$

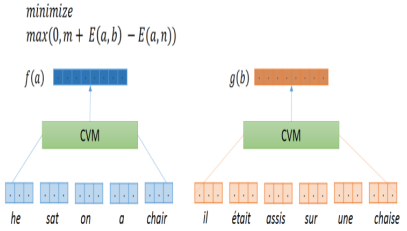
$$-corr(h(\bar{X}), h(\bar{Y}))$$

Lets compare the performance of some of these approaches on the task of cross language document classification

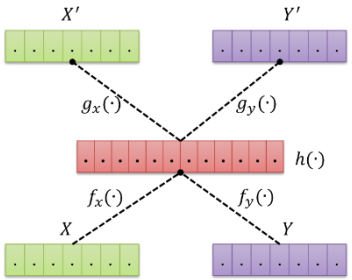




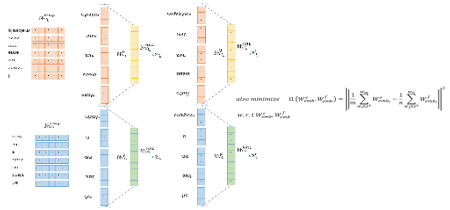
(Klementiev et. al., 2012)



(Herman & Blunsom, 2014)



(Chandar et.al., 2014)



(Gouws et. al., 2015)

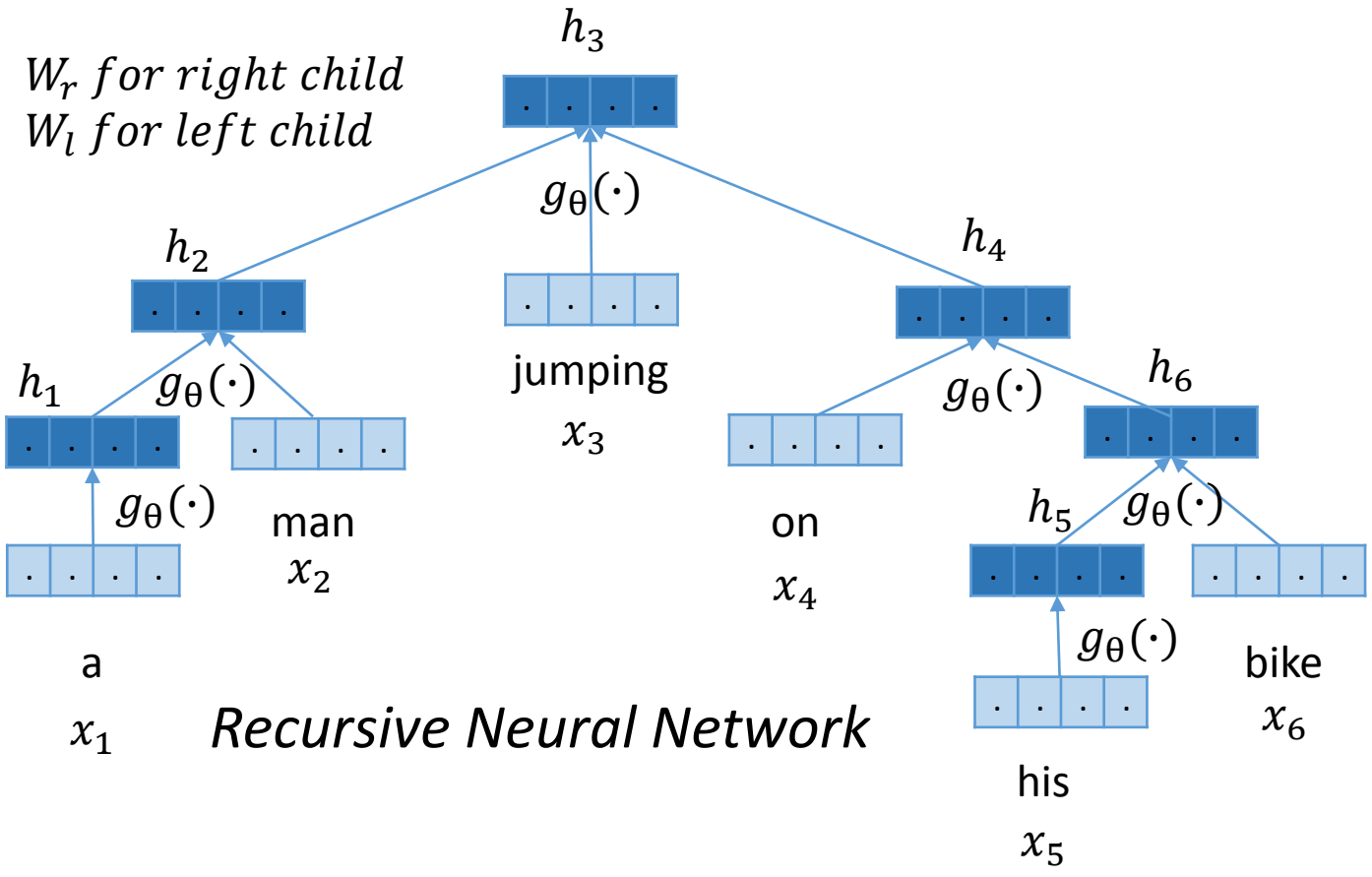
Approach	en→de	de → en
Klementiev et. al., 2012	77.6	71.1
Herman & Blunsom, 2013	83.7	71.4
Chandar et.al., 2014	**91.8**	72.8
Gouws et. al., 2015	86.5	**75.0**

We now look at multimodal representation learning

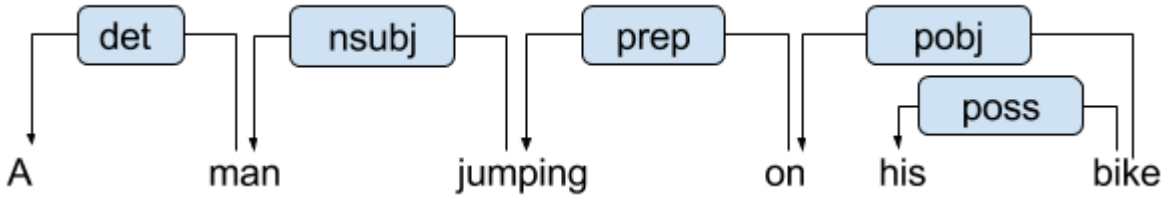
$$h_1 = g_\theta(x_1) = f(W_v x_1)$$

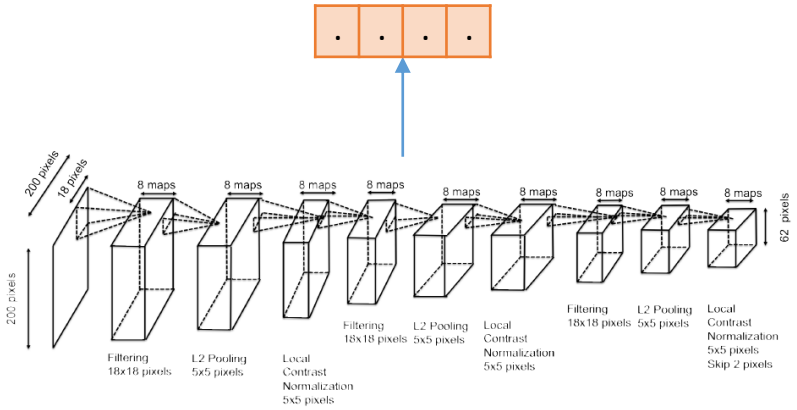
$$h_6 = g_\theta(x_6, h_5) = f(W_v x_6 + W_l h_5)$$

$$h_3 = g_\theta(x_3, h_2, h_4) = f(W_v x_6 + W_l h_2 + W_r h_4)$$



Recursive Neural Network





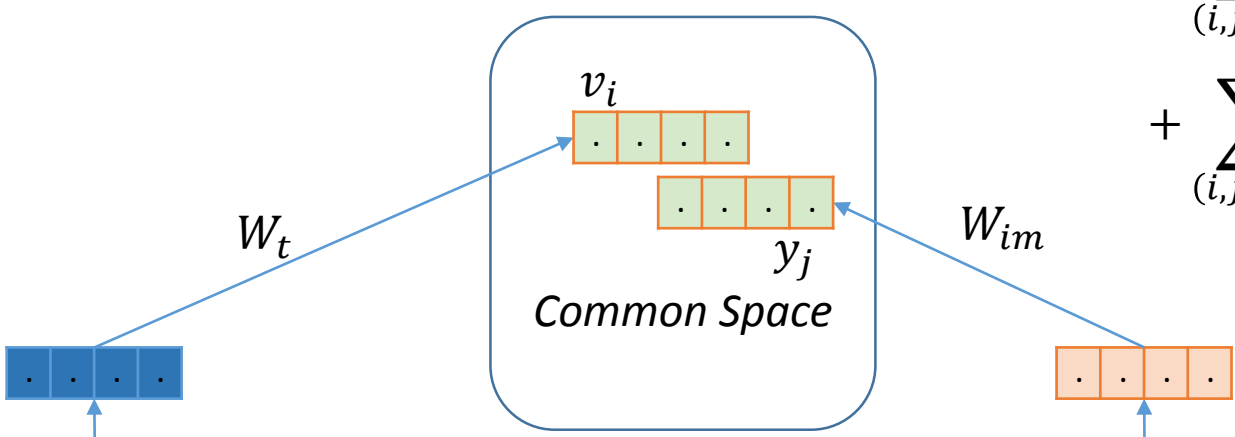
Convolutional Neural Network

$$Objective: \sum_{(i,j)\in P} \sum_{c\in S\setminus S(i)} \max(0, \Delta - v_i^T y_j + v_i^T y_c)$$

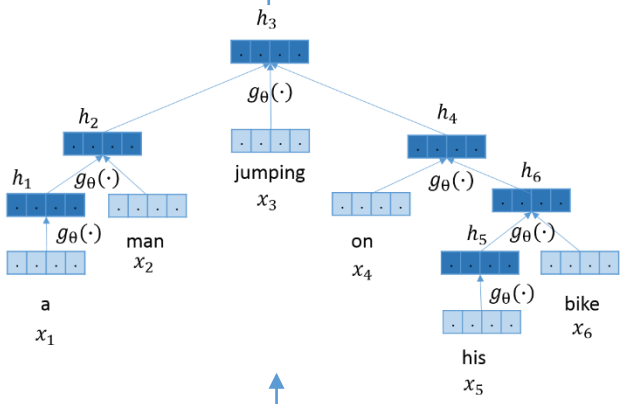
$$+ \sum_{(i,j)\in P} \sum_{c\in I\setminus I(j)} \max(0, \Delta - v_i^T y_j + v_i^T y_c)$$

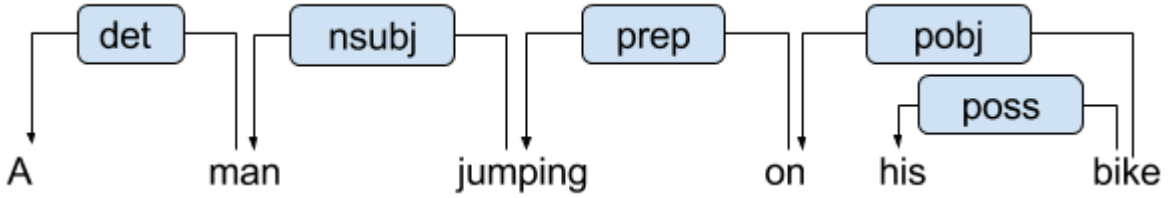
$(i,j) = correct\ pair$

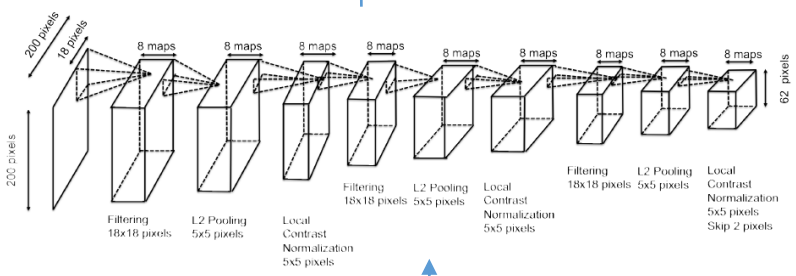
$(i,c) = incorrect\ pair$



Recursive Neural Network







Convolutional Neural Network

Interested in more ?

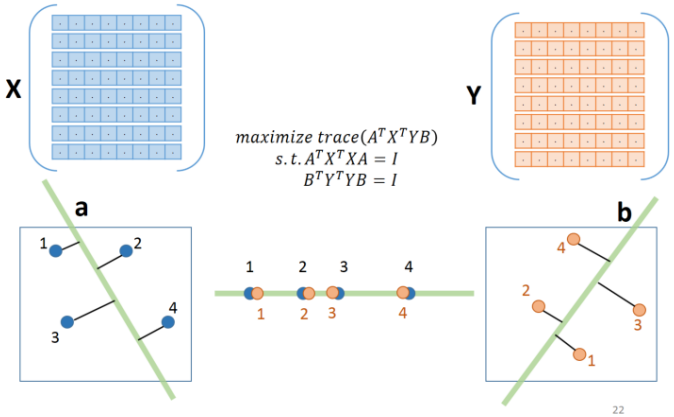
Bridge Correlational Neural Networks

@ NAACL, 2016

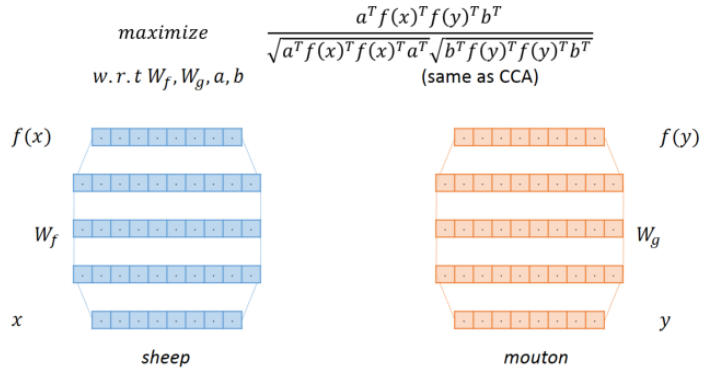
Monday, June 13, 2016

2:40 – 3:00 p.m.

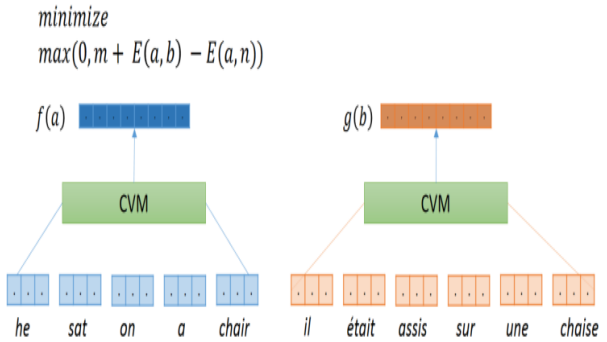
A quick summary … … …



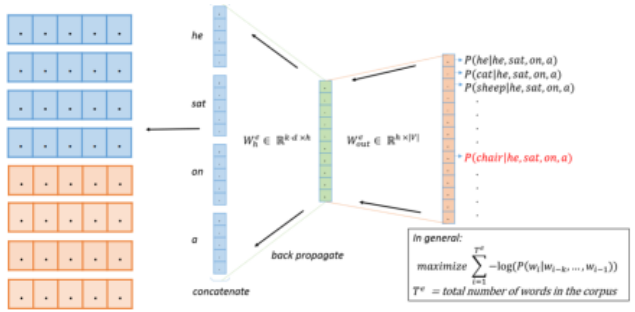
(Faruqui and Dyer, 2014)



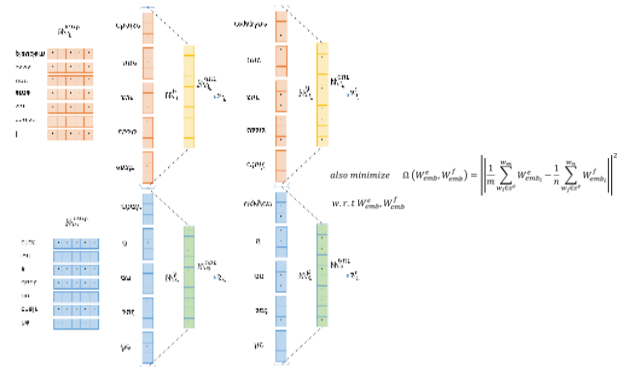
(Lu et. al., 2015)



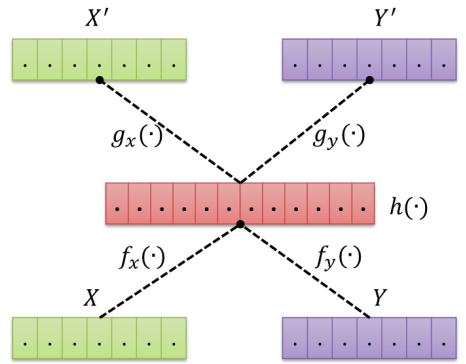
(Herman & Blunsom, 2014)



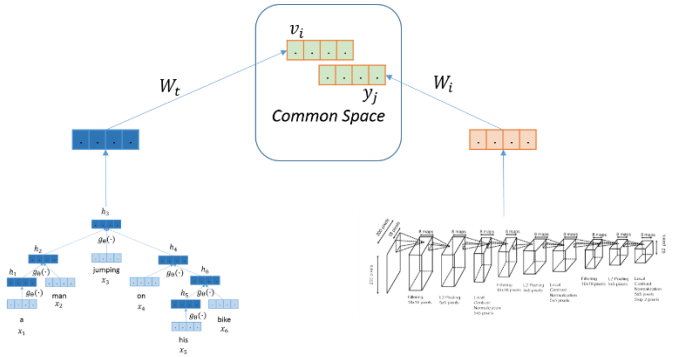
(Klementiev et. al., 2012)



(Gouws et. al., 2015)



(Chandar et. al., 2015)



(Socher et. al., 2013)

Research Directions: Representation Learning

- Learn task specific bilingual embeddings

$$maximize \sum_{j \in \{e,f\}} \sum_{i=1}^{T_j} \mathcal{L}(\theta^j) + \lambda \cdot \Omega\left(W^e_{emb}, W^f_{emb}\right) + \mathcal{L}_{task}(\alpha)$$

$w.r.t\ \theta_e, \theta_f, \alpha$

$\theta_e = W_e, W^e_h, W^e_{out}$

$\theta_f = W_f, W^f_h, W^f_{out}$

$\alpha =$ *task –specific parameters*

monolingual similarity *bilingual similarity*

- Learn from comparable corpora (instead of parallel corpora)
- Handle data imbalance
  - More data for $\mathcal{L}(\theta^j)$
  - Less data for $\lambda \cdot \Omega\left(W^e_{emb}, W^f_{emb}\right)$
- Handle larger vocabulary

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and open problems

Natural Language Generation

- Natural Language Processing
  - Natural Language Understanding (NLU)
  - Natural Language Generation (NLG)
- Natural language generation is hard.
- Applications of NLG:
  - Machine Translation
  - Question answering
  - Captioning
  - Summarization
  - Dialogue systems
- Evaluating NLG systems is also hard! (More about in the end)

Multilingual Multimodal NLG

Multilingual Multimodal NLG refers to conditional NLG where the generator could be conditioned on

- Multiple languages (machine translation, summarization, dialog systems)
- Multiple modalities like images, videos (visual QA, image captioning, video captioning)

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and research directions

Machine Translation

En: Economic growth has slowed down in recent years.

Fr : La croissance économique a ralenti au cours des dernières années .

- Statistical Machine Translation (SMT) aims to design systems that can learn to translate between languages based on some training data.
- SMT maximizes

$$p(f|e) \propto p(e|f)p(f)$$

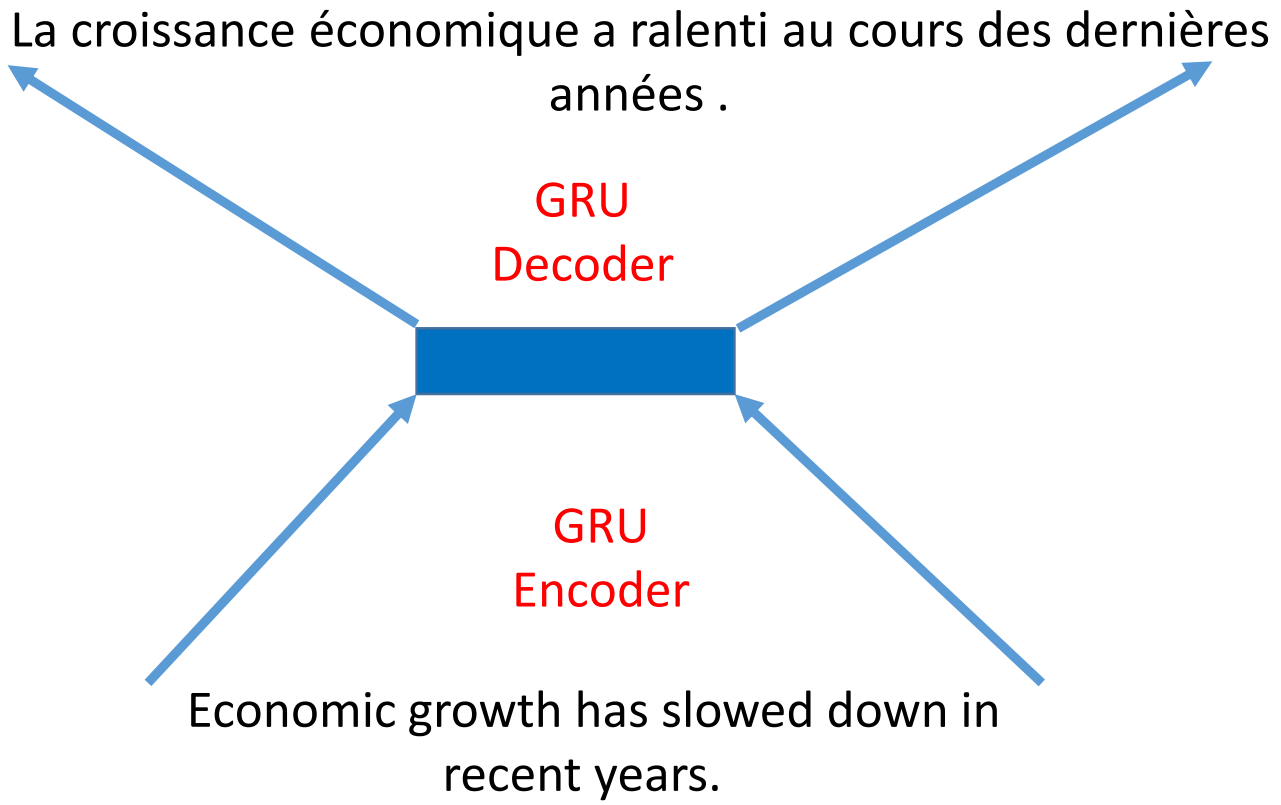
- p(e|f) – translation model
- p(f) – language model
- Traditional methods – long pipeline (Example: Moses, Joshua)

Neural Machine Translation

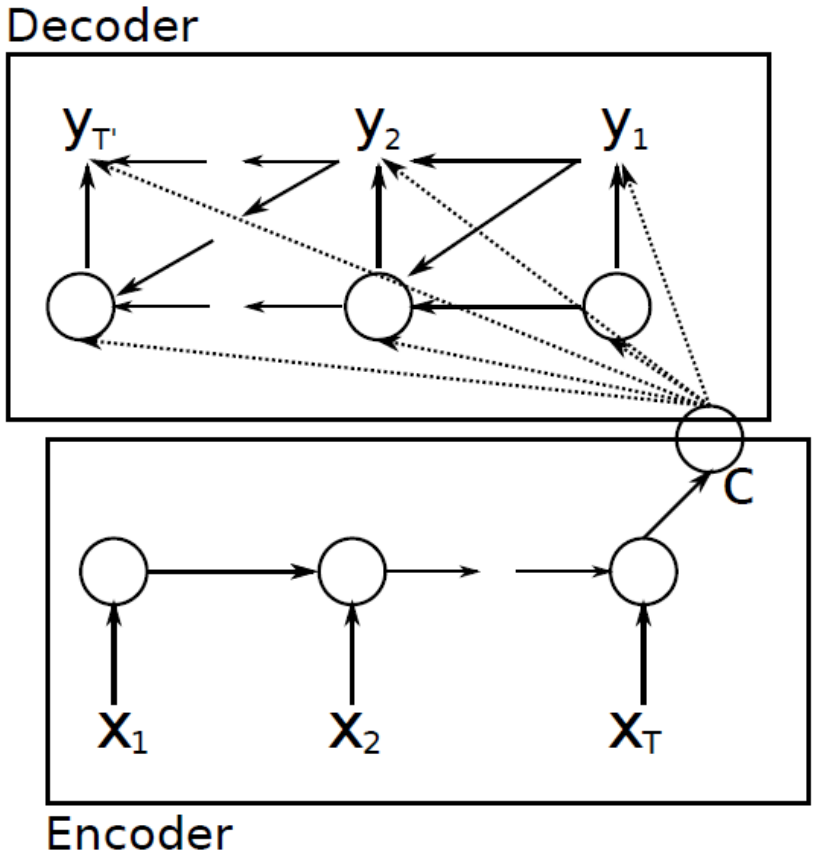
- Neural Network based machine translation system.
- Why Neural MT?
  - Easy to train in an end-to-end fashion.
  - The whole system can be optimized for the actual task in hand.
  - No need to store gigantic phrase tables. Small memory footprint.
  - Simple decoder unlike highly intricate decoders in standard MT.
- We will consider
  - Single source – single target NMT
  - Multi source – single target NMT
  - Single source – multi target NMT
  - Multi source – multi target NMT

Learning phrase representations using RNN Encoder-Decoder for SMT (Cho et al., 2014)

- RNN Encoder: encode a variable-length sequence into a fixed-length vector representation.
- RNN Decoder: decode a given fixed-length vector representation back into a variable-length sequence.



RNN Encoder-Decoder



$$X_n = x_1, ..., x_T$$

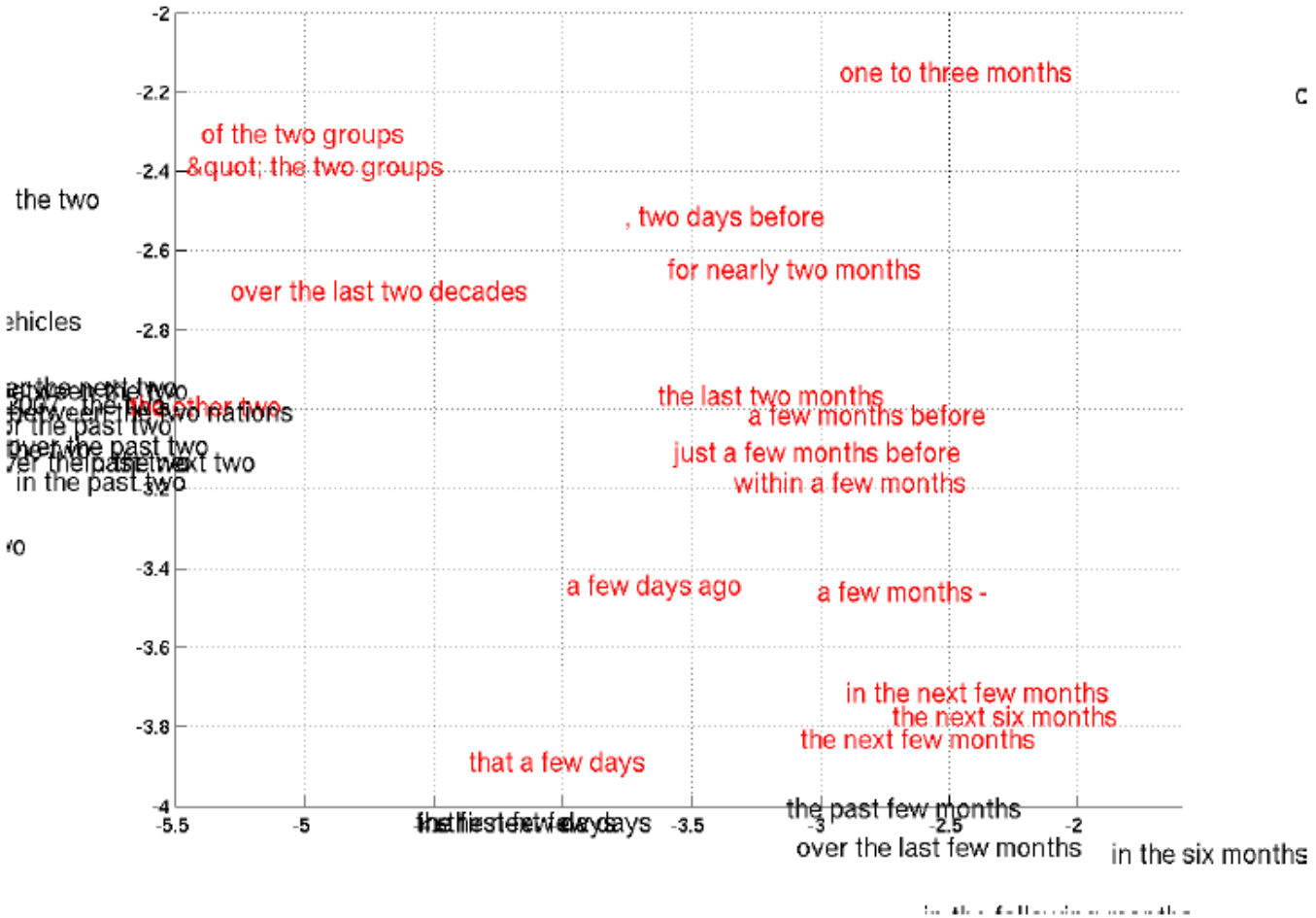
$$Y_n = y_1, ..., y_{T'}$$

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log\, p_\theta(Y_n | X_n)$$

How to use the trained model?

1. Generate a target sequence given an input sequence.
2. Score a given pair of input/output sequence p(y|x).

2-D embedding of the learned phrase representation

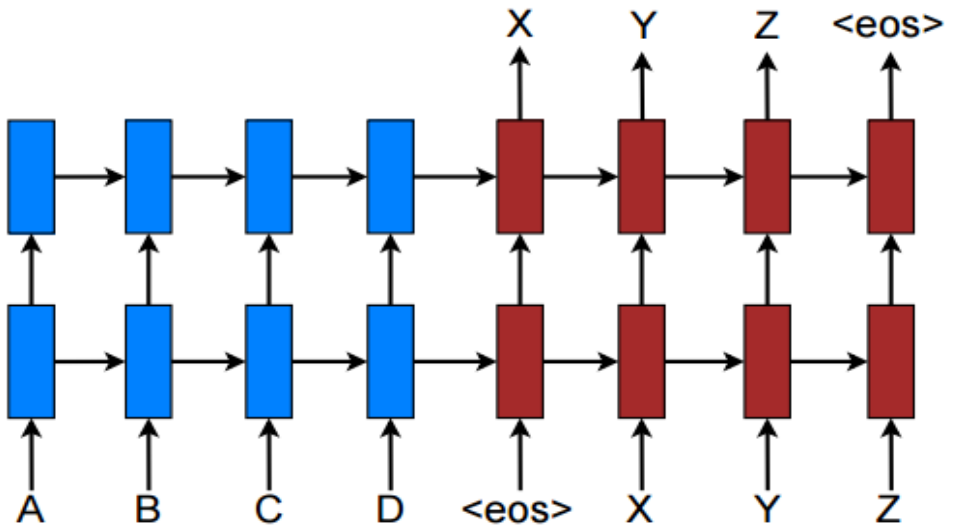


WMT'14 English/French SMT - rescoring

Baseline: Moses with default setting

Models	BLEU	
	dev	test
Baseline	30.64	33.30
RNN	31.20	33.87
CSLM + RNN	31.48	34.64
CSLM + RNN + WP	31.50	34.54

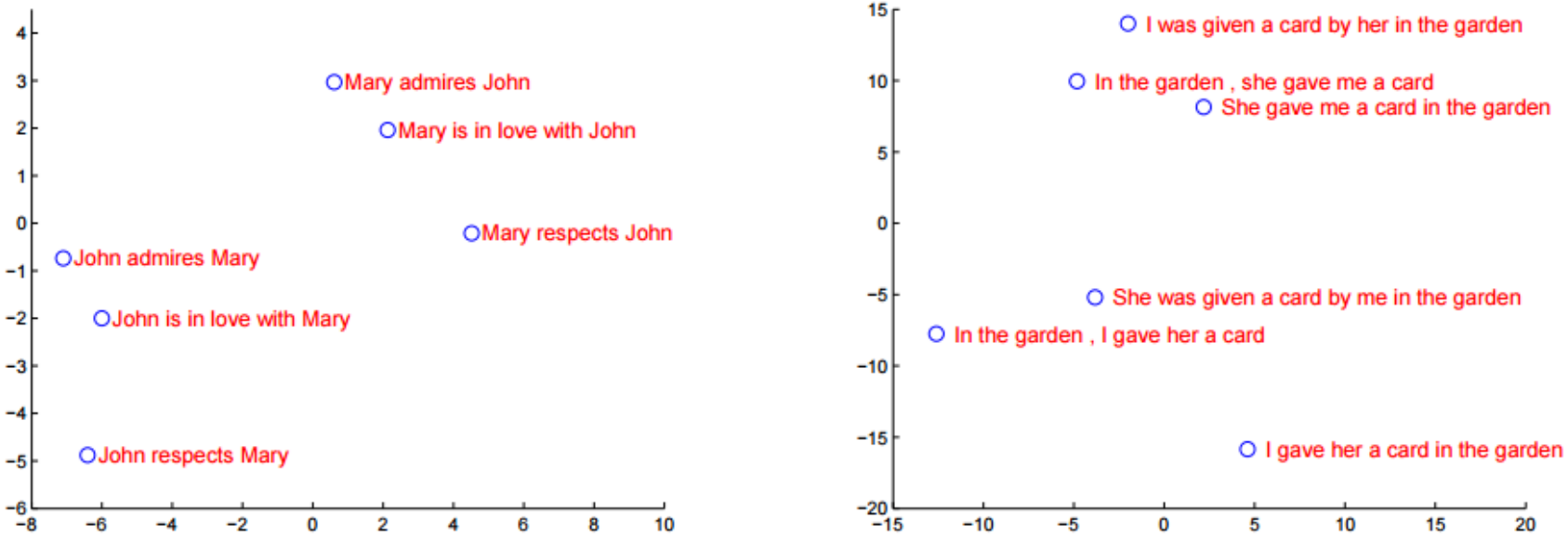
Sequence to Sequence Learning with Neural Networks (Sutskever et al., 2014)



What is different from Cho et al., 2014?

1. LSTM encoder/decoder instead of GRU encoder/decoder.
2. 4 layers deep encoder/decoder instead of shallow encoder/decoder.

2-D embedding of the learned phrase representation



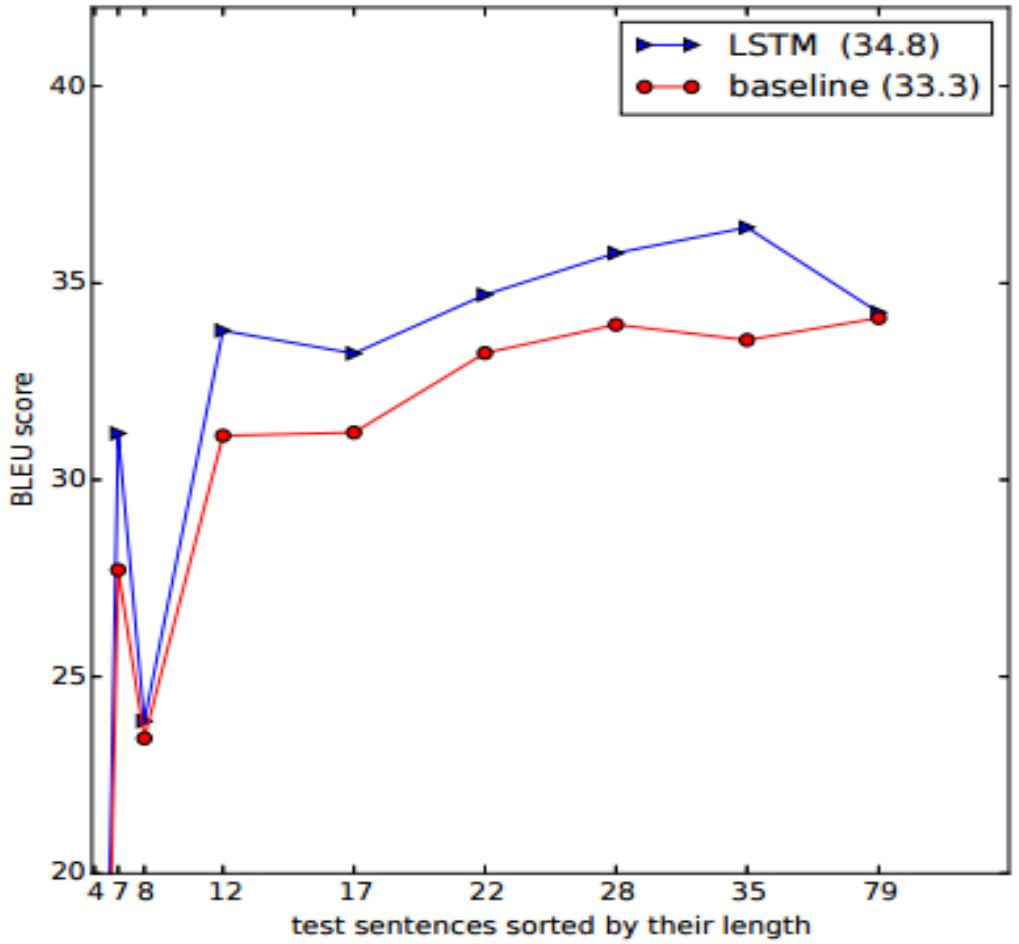
The performance of the LSTM on WMT'14 English to French test set

Trick-1: Reverse the input sequence.

Method	test BLEU score (ntst14)
Bahdanau et al. [2]	28.45
Baseline System [29]	33.30
Single forward LSTM, beam size 12	26.17
Single reversed LSTM, beam size 12	30.59
Ensemble of 5 reversed LSTMs, beam size 1	33.00
Ensemble of 2 reversed LSTMs, beam size 12	33.27
Ensemble of 5 reversed LSTMs, beam size 2	34.50
Ensemble of 5 reversed LSTMs, beam size 12	**34.81**

Trick-2: Ensemble Neural Nets.

Performance as a function of sentence length

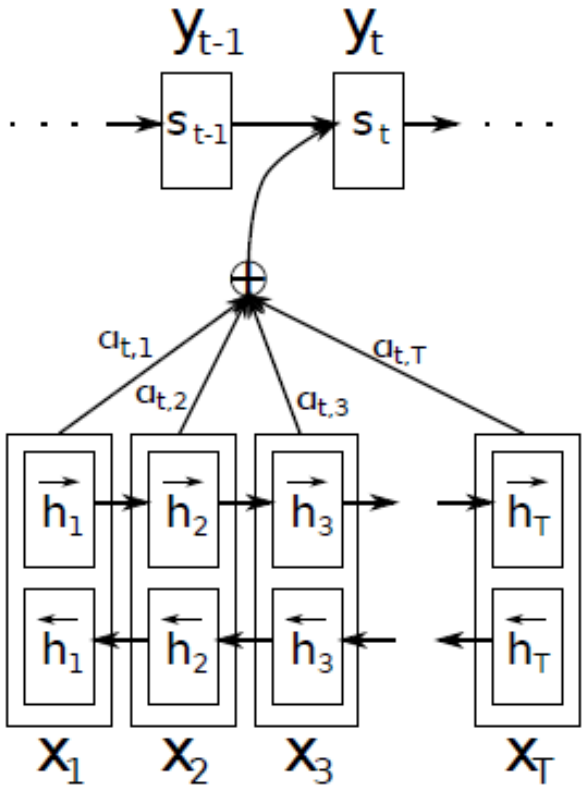


Neural MT by jointly learning to align and translate (Bahdanau et al., 2015)

- Issue with encoder-decoder approach
  - Can we compress all the necessary information in a sentence to a fixed length vector?

    NO
- How to choose the length of the vector?
  - It should be proportional to the length of the sentence.
- Bahdanau et al. proposed to use *k* fixed length vectors for encoder where *k* is the length of the sentence.

Attention based NMT

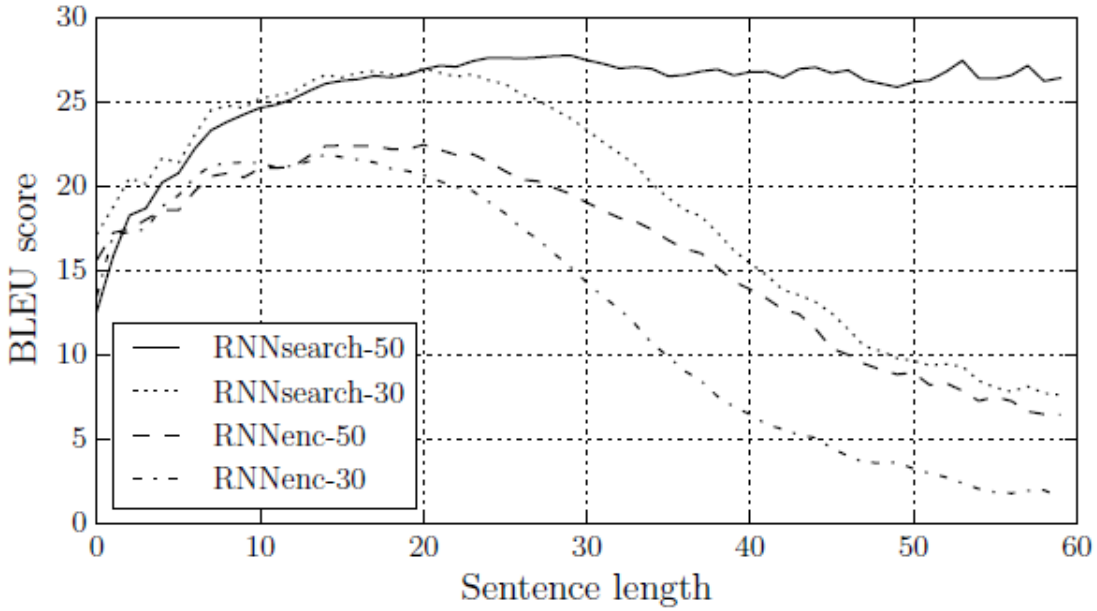


- To generate t^{th} word
  - Model learns an attention mechanism over all the words in the input sentence.
- Input words are represented using a bi-directional RNN.

Results on WMT'14 En-Fr Dataset

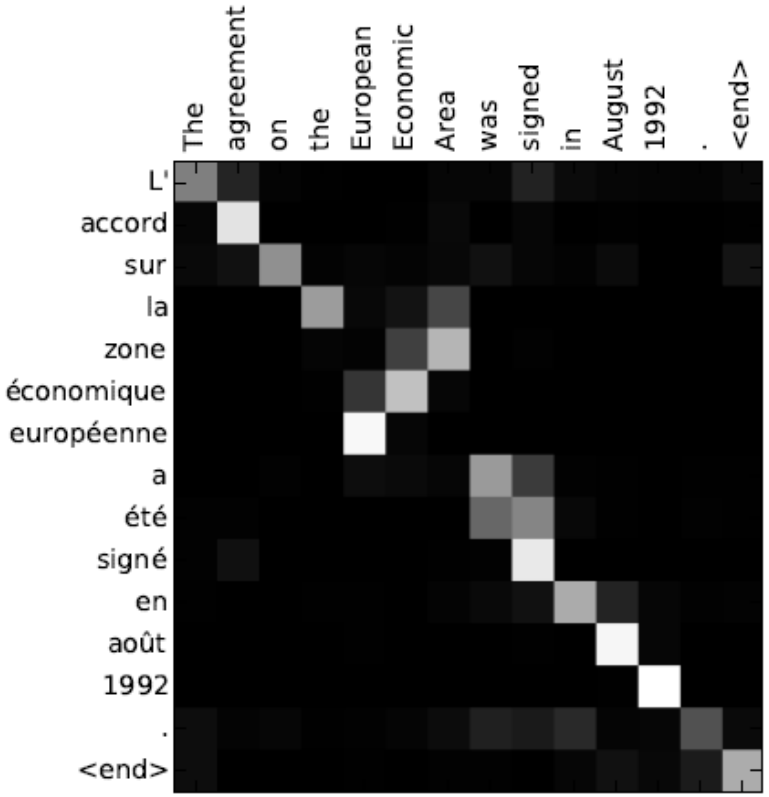
Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

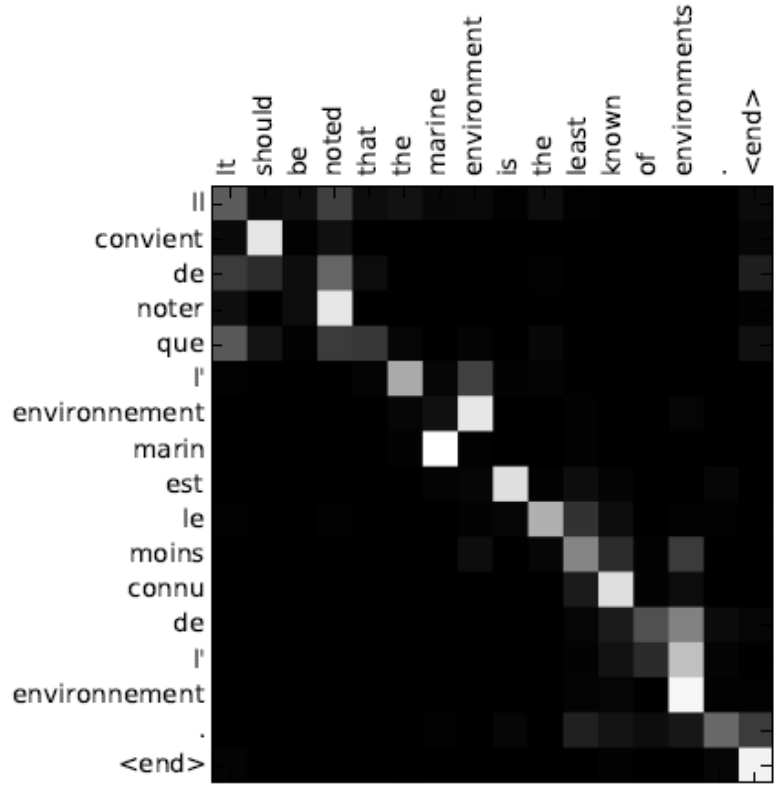
BLEU score



Effect of increasing the sentence length

Visualization of attention



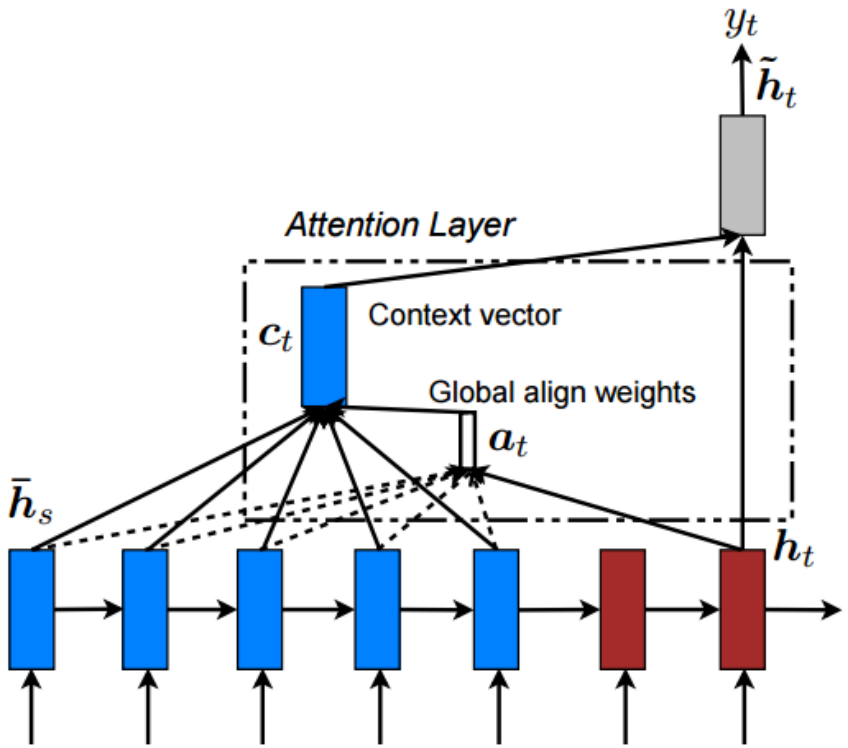


Effective Approaches to Attention-based NMT

(Luong et al., 2015)

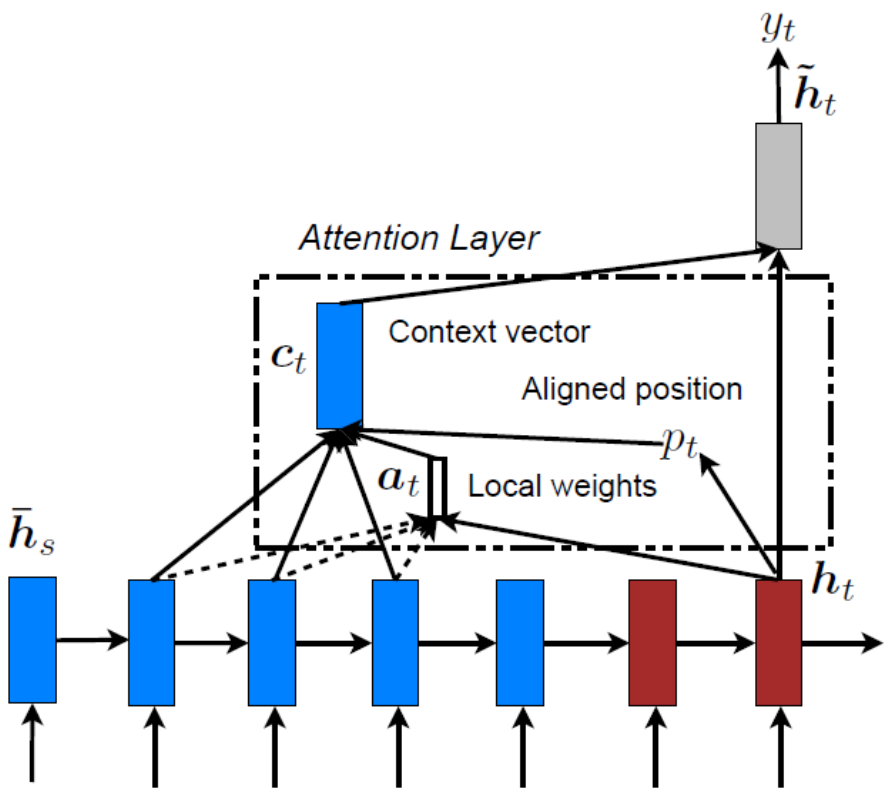
- Exploring various architectural choices for attention based NMT systems.
  - Global Attention
  - Local Attention
  - Input-feeding approach

Global Attention



- Stacked LSTM encoder instead of bi-directional single layer LSTM encoder.
- Hidden state h_t of the final layer is used for context computation.
- All the source words are used for computing the context (similar to Bahdanau et al.)

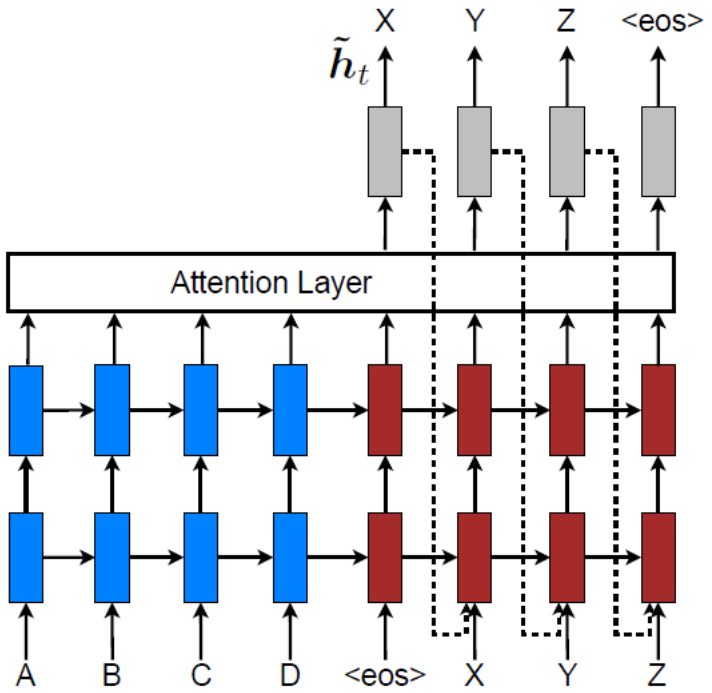
Local Attention



- The model first predicts a single aligned position p_t for the current target word.
- A window chosen around p_t is used to compute the context c_t
- c_t is a weighted average of the source hidden states in the window.

- In between soft attention and hard attention (more on hard attention later).

Input-feeding Approach



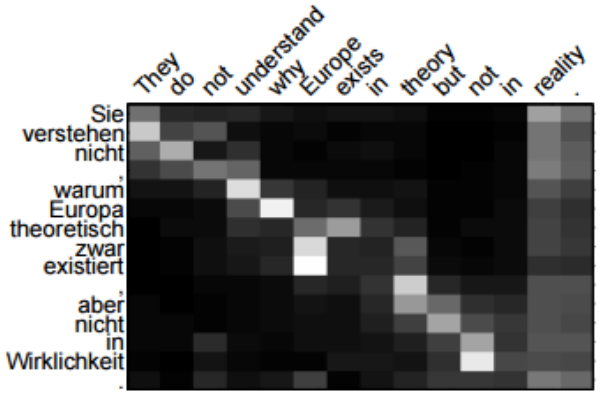
- Attention vectors are fed as input to the next time steps to inform the model.
- Similar to *coverage* set in standard MT.

WMT'14 En-De results

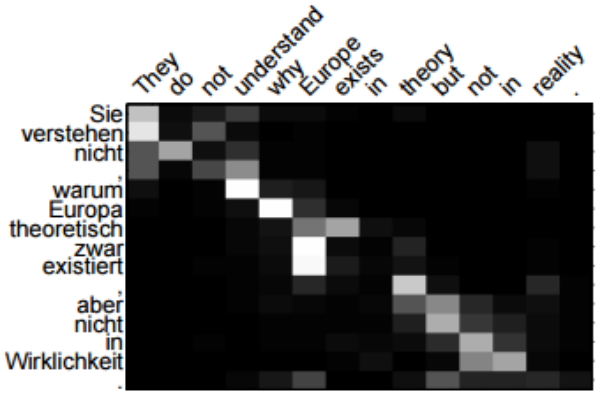
System	Ppl	BLEU
Winning WMT'14 system – *phrase-based* + *large LM* (Buck et al., 2014)		20.7
Existing NMT systems		
RNNsearch (Jean et al., 2015)		16.5
RNNsearch + unk replace (Jean et al., 2015)		19.0
RNNsearch + unk replace + large vocab + *ensemble* 8 models (Jean et al., 2015)		**21.6**
Our NMT systems		
Base	10.6	11.3
Base + reverse	9.9	12.6 (+*1.3*)
Base + reverse + dropout	8.1	14.0 (+*1.4*)
Base + reverse + dropout + global attention (*location*)	7.3	16.8 (+*2.8*)
Base + reverse + dropout + global attention (*location*) + feed input	6.4	18.1 (+*1.3*)
Base + reverse + dropout + local-p attention (*general*) + feed input	5.9	19.0 (+*0.9*)
Base + reverse + dropout + local-p attention (*general*) + feed input + unk replace		20.9 (+*1.9*)
Ensemble 8 models + unk replace		**23.0** (+*2.1*)

Alignment visualizations

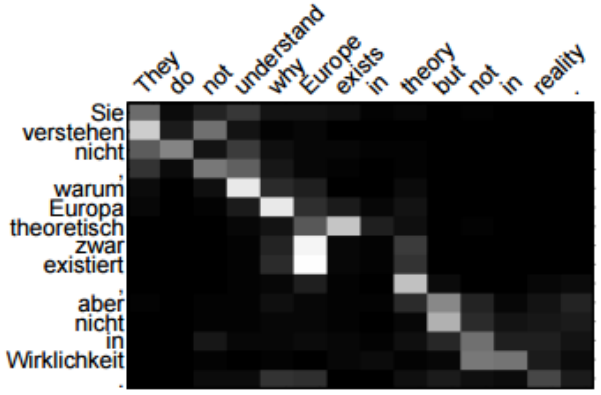
Global attention



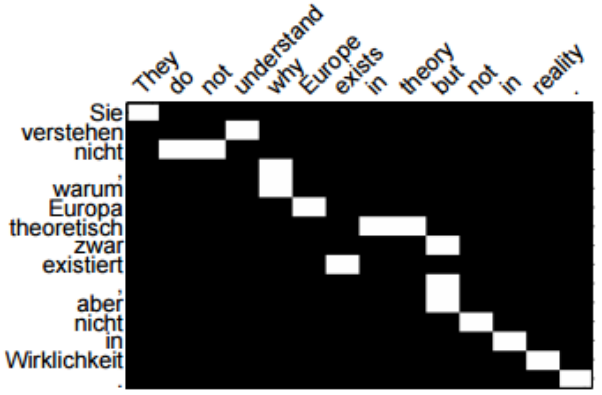
Local-m attention



Local-p attention



Gold alignment



Alignment Error Rate on RWTH En-De alignment data

Method	AER
global (location)	0.39
local-m (general)	0.34
local-p (general)	0.36
ensemble	0.34
Berkeley Aligner	0.32

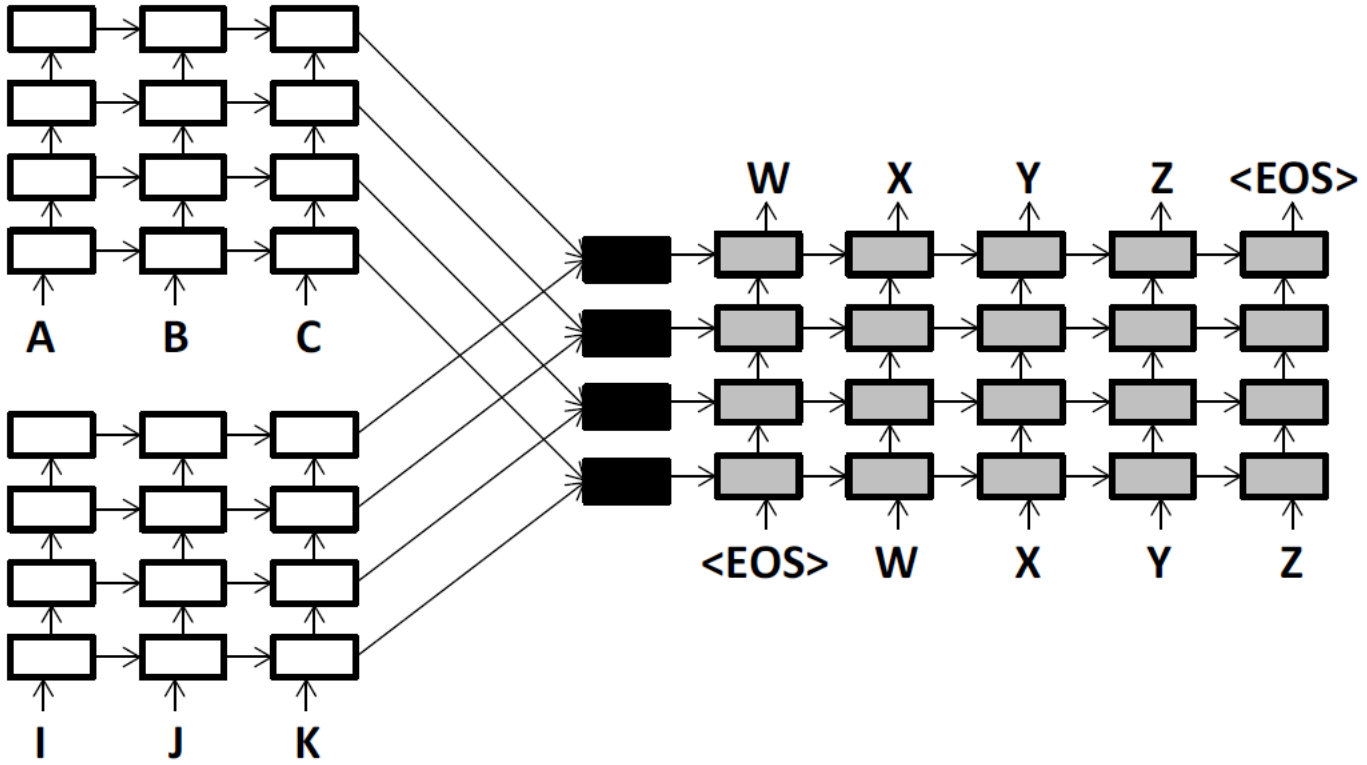
Multi-source NMT

- Why multiple sources?
- Two strings can reduce ambiguity via *triangulation*.
- Ex: English word "bank" may be easily translated to French in the presence of a second, German input containing the work "Flussufer" (river bank).
- Sources should be distant languages for better triangulation.
  - ✓ English and German to French
  - × English and French to German

Multi-source NMT (Zoph and Knight, 2016)

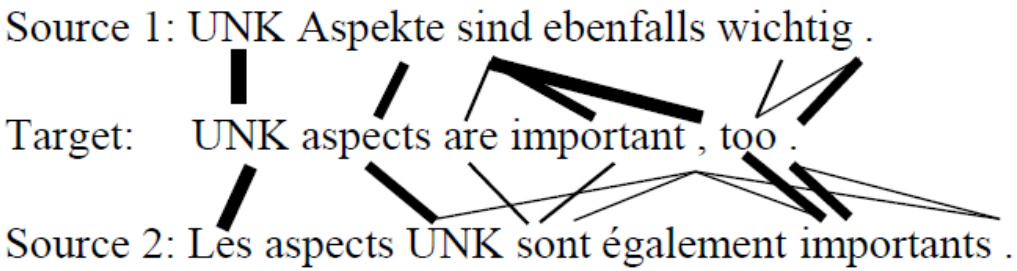
- Train p(e|f, g) model directly on trilingual data.
- Use it to decode e given any (f, g) pair.
- How to combine information from f and g?

Multi-source Encoder-Decoder Model



Multi-source Attention model

- We can take local-attention NMT model and concatenate context from multiple sources

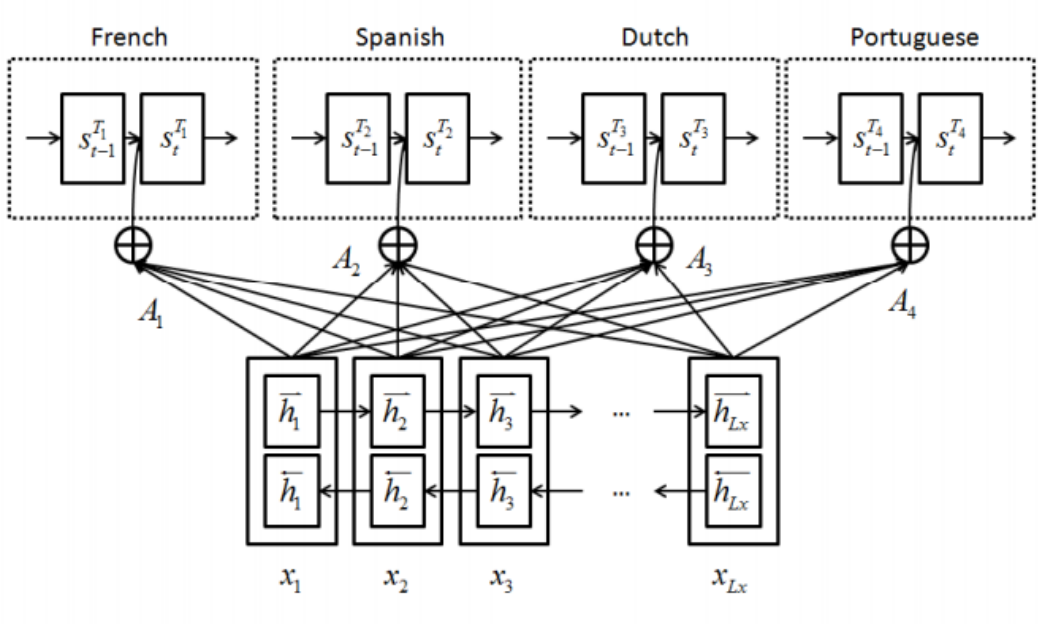


English-French-German NMT

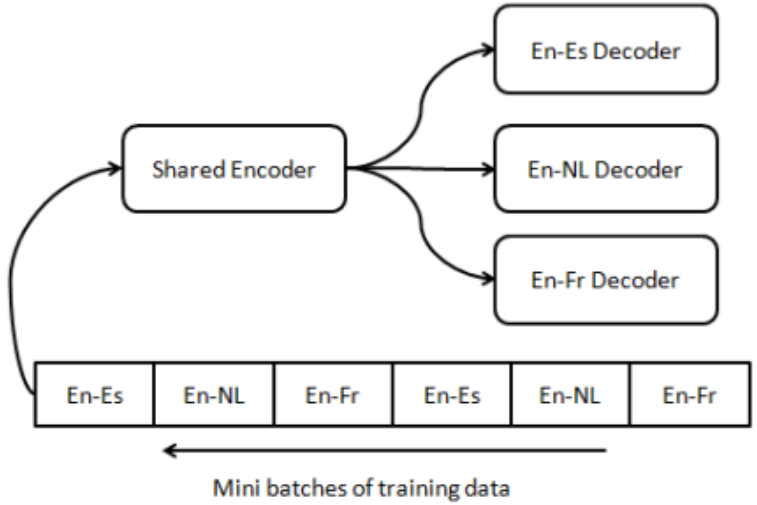
Target = English			
Source	**Method**	**Ppl**	**BLEU**
French	—	10.3	21.0
German	—	15.9	17.3
French+German	Basic	8.7	23.2
French+German	Child-Sum	9.0	22.5
French+French	Child-Sum	10.9	20.7
French	Attention	8.1	25.2
French+German	B-Attent.	5.7	30.0
French+German	CS-Attent.	6.0	29.6

Target = German			
Source	**Method**	**Ppl**	**BLEU**
French	—	12.3	10.6
English	—	9.6	13.4
French+English	Basic	9.1	14.5
French+English	Child-Sum	9.5	14.4
English	Attention	7.3	17.6
French+English	B-Attent.	6.9	18.6
French+English	CS-Attent.	7.1	18.2

Multi-Target NMT (Dong et alt., 2015)



Multi-task learning framework for multiple-target language translation



Optimization for end to multi-end model

Multi-Target NMT (Dong et alt., 2015)

Training Data Information						
Lang	En-Es	En-Fr	En-Nl	En-Pt	En-Nl-sub	En-Pt-sub
Sent size	1,965,734	2,007,723	1,997,775	1,960,407	300,000	300,000
Src tokens	49,158,635	50,263,003	49,533,217	49,283,373	8,362,323	8,260,690
Trg tokens	51,622,215	52,525,000	50,661,711	54,996,139	8,590,245	8,334,454

Size of training corpus for different language pairs

	Nmt Baseline	Nmt Multi-Full	Nmt Multi-Partial	Moses
En-Fr	23.89	26.02(**+2.13**)	25.01(**+1.12**)	23.83
En-Es	23.28	25.31(**+2.03**)	25.83(**+2.55**)	23.58

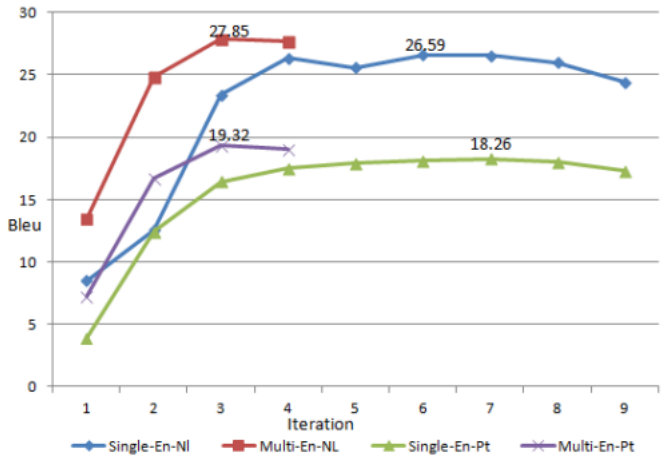
Multi-task NMT v.s. single model v.s. moses on the WMT 2013 test set

Lang-Pair	En-Es	En-Fr	En-Nl	En-Pt
Single NMT	26.65	21.22	28.75	20.27
Multi Task	28.03	22.47	29.88	20.75
Delta	**+1.38**	**+1.25**	**+1.13**	**+0.48**

Multi-task neural translation v.s. single model given large-scale corpus in all language pairs

Lang-Pair	En-Es	En-Fr	En-Nl*	En-Pt*
Single NMT	26.65	21.22	26.59	18.26
Multi Task	28.29	21.89	27.85	19.32
Delta	**+1.64**	**+0.67**	**+1.26**	**+1.06**

Multi-task neural translation v.s. single model with a small-scale training corpus on some language pairs. * means that the language pair is sub-sampled.



Faster and Better convergence in Multi-task Learning in multiple language translation

Multi-Way, Multilingual NMT (Firat et alt., 2016)

- Multiple sources and multiple targets
- Advantage?
  - Sharing knowledge across multiple languages.
  - One universal common space for multiple languages.
  - Advantageous for low resource languages?
- Challenges
  - N-lingual data is difficult to get for N>2.
  - Parameters of the system grows fast w.r.t number of languages.

Simple Solutions

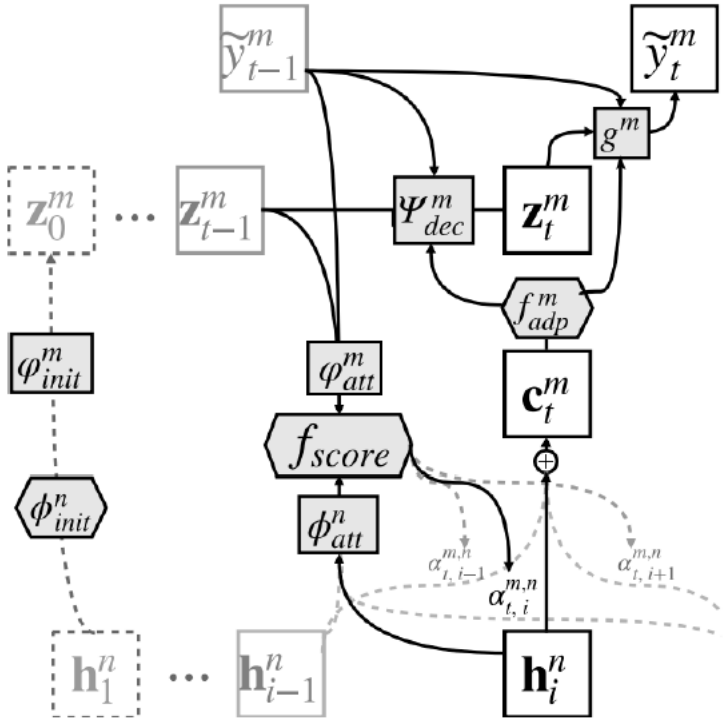
- Encoder-decoder model with multiple encoders and multiple decoders which are shared across language pairs.
- Can we do the same method with attention based models?
  - Attention is language pair specific.
  - With L languages, we need $O(L^2)$ attentions.
  - Can we share same attention module to multiple languages?

    YES

Multi-way Multilingual NMT (Firat et al., 2016)

- Model
  - N encoders
  - N decoders
  - Shared attention mechanism
- Both encoders and decoders are shared across multiple language pairs.
- Each encoder can be of different type (convolutional/rnn, different sizes).

One step of multiway multilingual NMT



Low Resource Translation

	Size	Single	Single+DF	Multi
En→Fi	100k	5.06/3.96	4.98/3.99	6.2/**5.17**
	200k	7.1/6.16	7.21/6.17	8.84/**7.53**
	400k	9.11/7.85	9.31/8.18	11.09/**9.98**
	800k	11.08/9.96	11.59/10.15	12.73/**11.28**
De→En	210k	14.27/13.2	14.65/13.88	16.96/**16.26**
	420k	18.32/17.32	18.51/17.62	19.81/**19.63**
	840k	21/19.93	21.69/20.75	22.17/**21.93**
	1.68m	23.38/23.01	23.33/22.86	23.86/**23.52**
En→De	210k	11.44/11.57	11.71/11.16	12.63/**12.68**
	420k	14.28/14.25	14.88/15.05	15.01/**15.67**
	840k	17.09/17.44	17.21/17.88	17.33/**18.14**
	1.68m	19.09/19.6	19.36/20.13	19.23/**20.59**

Large Scale Translation

		Dir	Fr (39m)		Cs (12m)		De (4.2m)		Ru (2.3m)		Fi (2m)	
			→ En	En →	→ En	En →	→ En	En →	→ En	En →	→ En	En →
(a) BLEU	Dev	Single	27.22	26.91	21.24	15.9	24.13	20.49	21.04	18.06	13.15	9.59
		Multi	26.09	25.04	21.23	14.42	23.66	19.17	21.48	17.89	12.97	8.92
	Test	Single	27.94	**29.7**	20.32	**13.84**	24	**21.75**	22.44	**19.54**	12.24	**9.23**
		Multi	**28.06**	27.88	**20.57**	13.29	**24.20**	20.59	**23.44**	19.39	**12.61**	8.98
(b) LL	Dev	Single	-50.53	-53.38	-60.69	-69.56	-54.76	-61.21	-60.19	-65.81	-88.44	-91.75
		Multi	-50.6	-56.55	-54.46	-70.76	-54.14	-62.34	-54.09	-63.75	-74.84	-88.02
	Test	Single	-43.34	**-45.07**	-60.03	**-64.34**	-57.81	**-59.55**	-60.65	-60.29	-88.66	-94.23
		Multi	**-42.22**	-46.29	**-54.66**	-64.80	**-53.85**	-60.23	**-54.49**	**-58.63**	**-71.26**	**-88.09**

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and research directions

Image Captioning

Input

A women is throwing a frisbee in a park.

Output

Image Captioning

Input

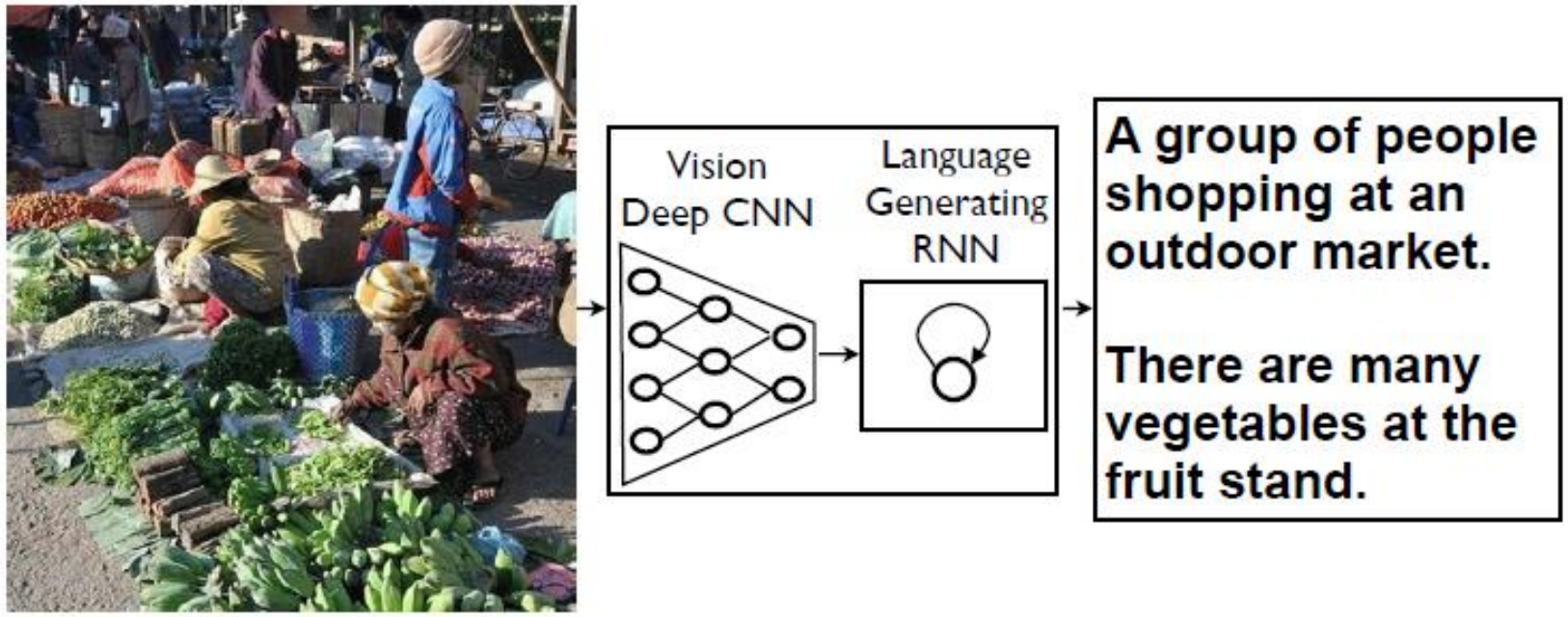
1. A building is surrounded by wide attractions, such as a horse statue and a statue of a giant hand and wrist, with a picnic table next to them.
2. A large hand statue outside of a country store.
3. A strange antique store with odd artwork outside near assorted tables and chairs.
4. A wooden shop with a large hand in the forecourt.
5. Country store has big hand with checkered-base statue and tables with benches on front yard

Crowdsourced Captions

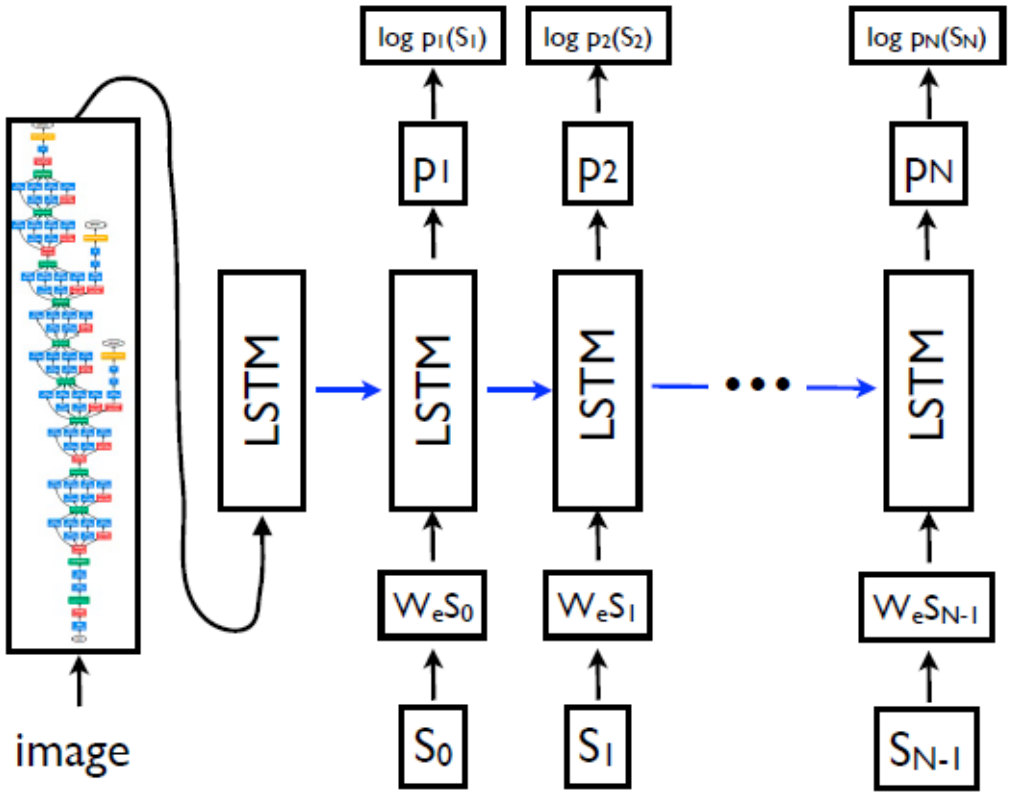
Image Captioning

- Requires both visual understanding and language understanding.
- Hard problem.
- Several potential applications.

Show and Tell: A Neural Image Caption Generator (Vinyals et al.,2015)



Show and Tell



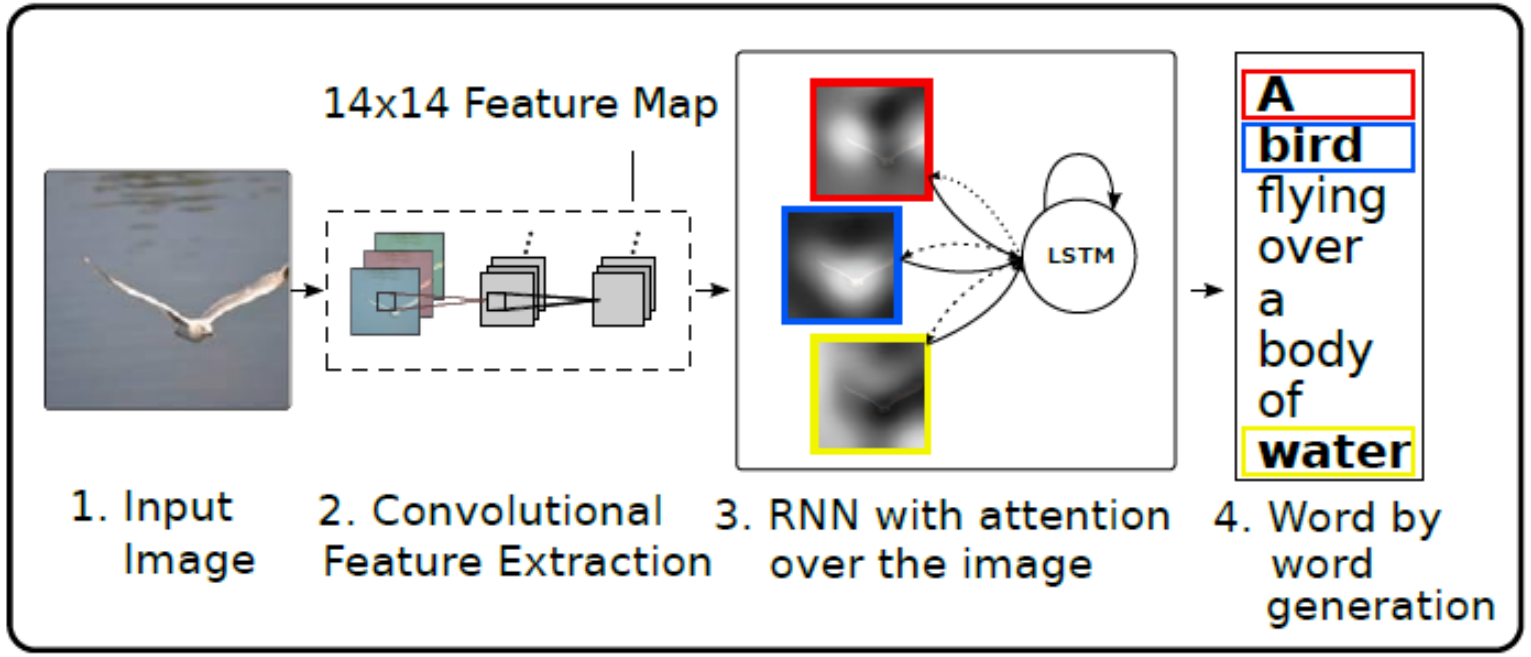
$$\log \ \ p(S|I) = \sum_{t=0}^{N} p(S_t|I, S_0, ..., S_{t-1})$$

Generated captions

BLEU-1 scores

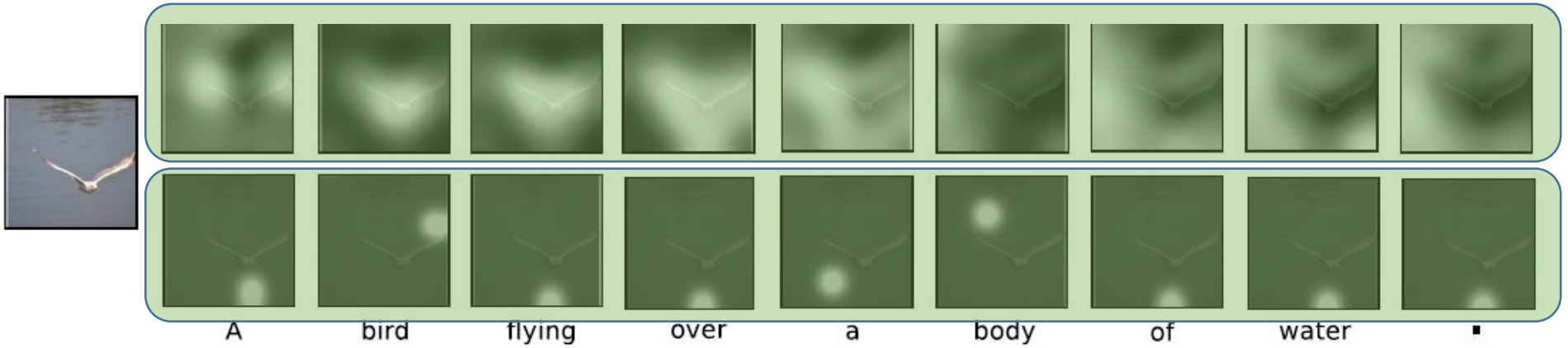
Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (Xu et al.,2016)



Attention over time

Soft attention



Hard attention

Show, Attend and Tell

- Show: Convolutional encoder

Instead of using final fully connected representation, use a lower convolutional layer.

This allows decoder to selective focus on certain parts of the image.

- Attend: Soft attention or hard attention over image (per time step)
- Tell: LSTM decoder conditioned on context at current time step, word generated in previous time step, previous hidden state.

Deterministic Soft Attention

- Same as the attention mechanism in Bahdanau et al.'s NMT system.

Given L vectors each corresponding to different parts of the image

$$a = \{a_1, ..., a_L\}, a_i \in \mathbb{R}^D$$

$$e_{ti} = f_{att}(a_i, h_{t-1})$$

$$\alpha_t = softmax(e_t)$$

$$\text{context} = \sum_{i=1}^{L} \alpha_{t,i} a_i$$

Stochastic Hard Attention

Given L vectors each corresponding to different parts of the image

$$a = \{a_1, ..., a_L\}, a_i \in \mathbb{R}^D$$

$$e_{ti} = f_{att}(a_i, h_{t-1})$$

$$\alpha_t = softmax(e_t)$$

Sample from α_t to select one part of the image.

$$s_t = Multinoulli_L(\{\alpha_i\})$$

S is a one-hot vector of size L.

This is not differentiable!

Use REINFORCE

$$\text{context} = \sum_{i=1}^{L} s_{t,i} a_i$$

Attending correct objects

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

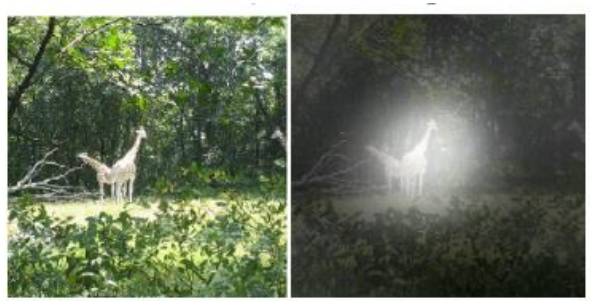
A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Examples of mistakes



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.

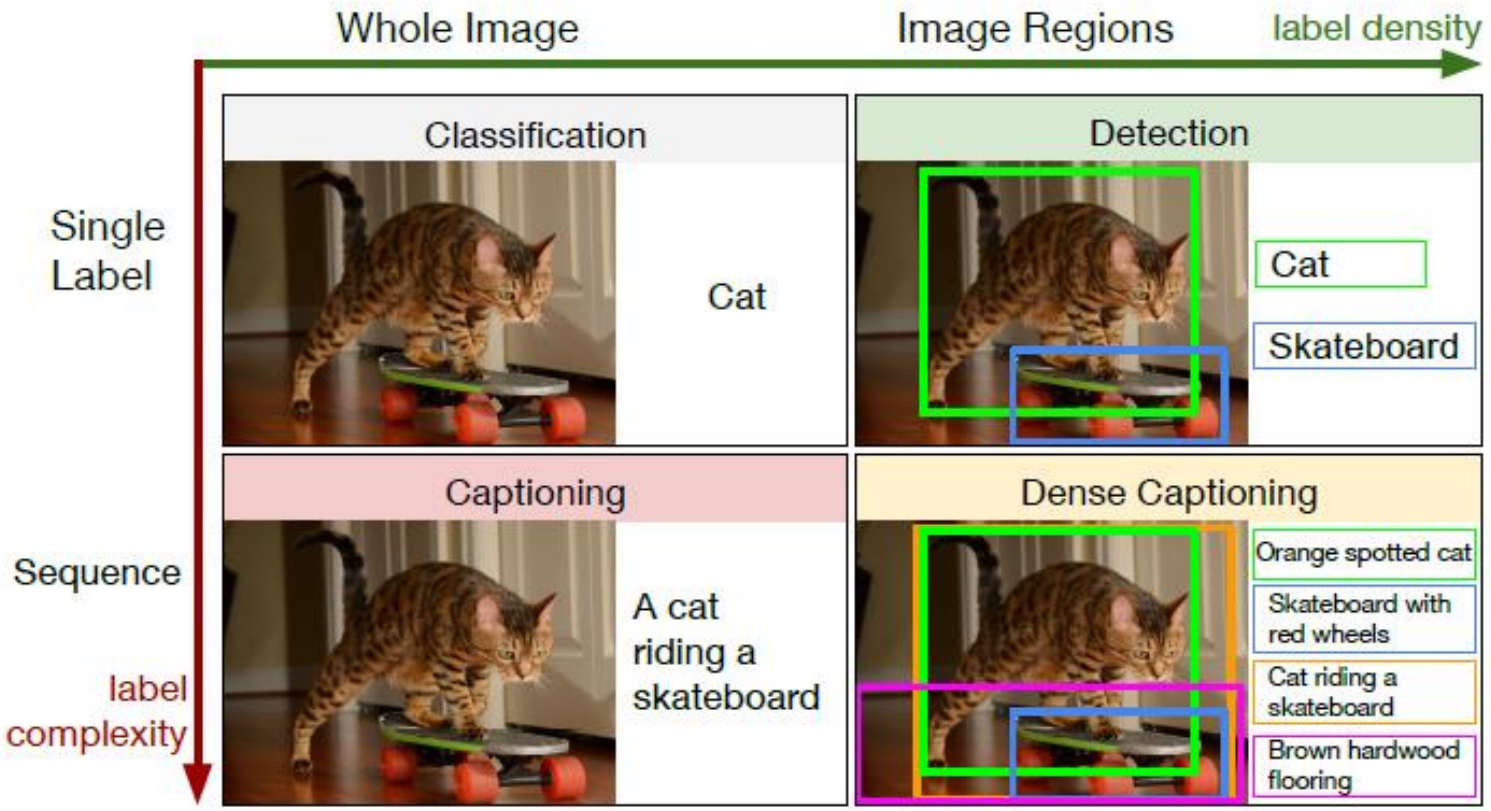
A woman is sitting at a table with a large pizza.

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

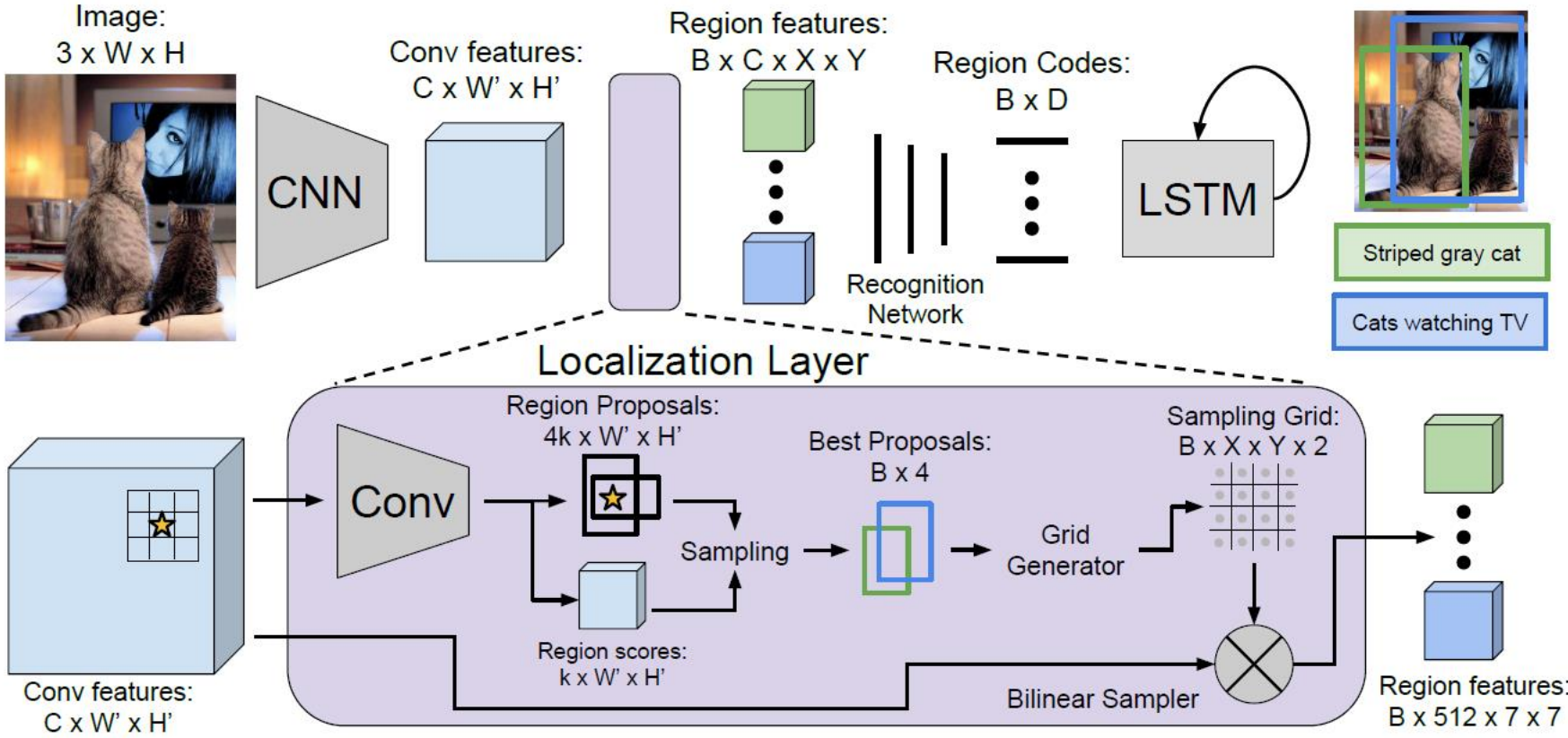
Results of 3 datasets

		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

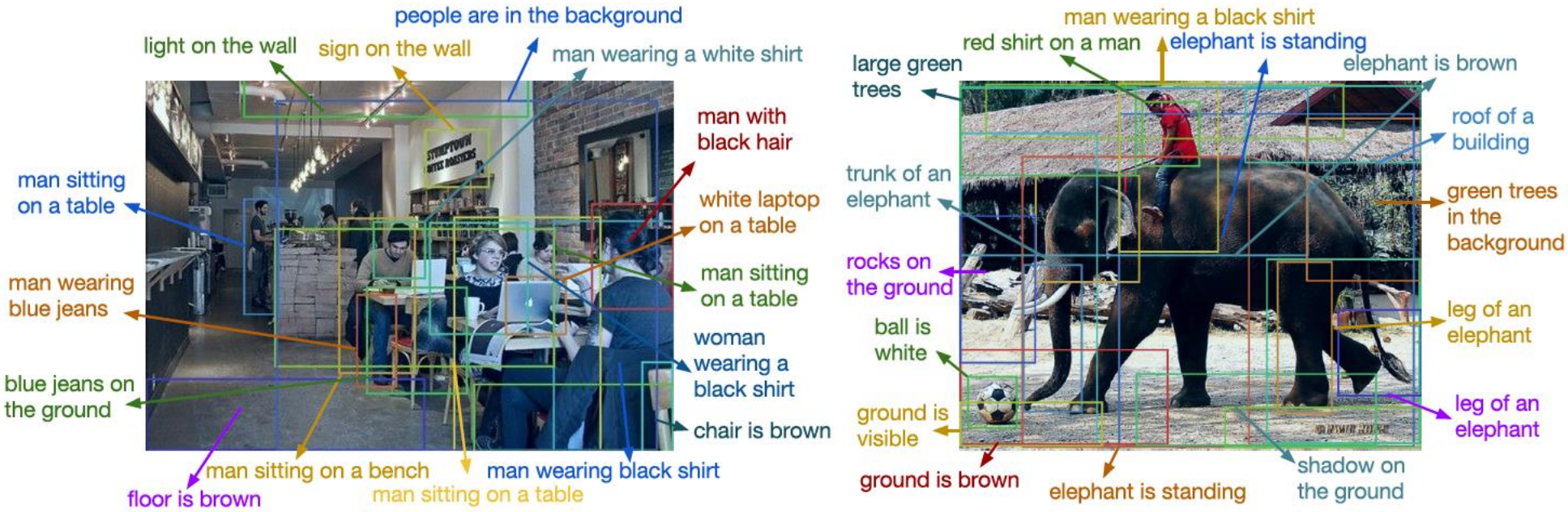
Dense Captioning



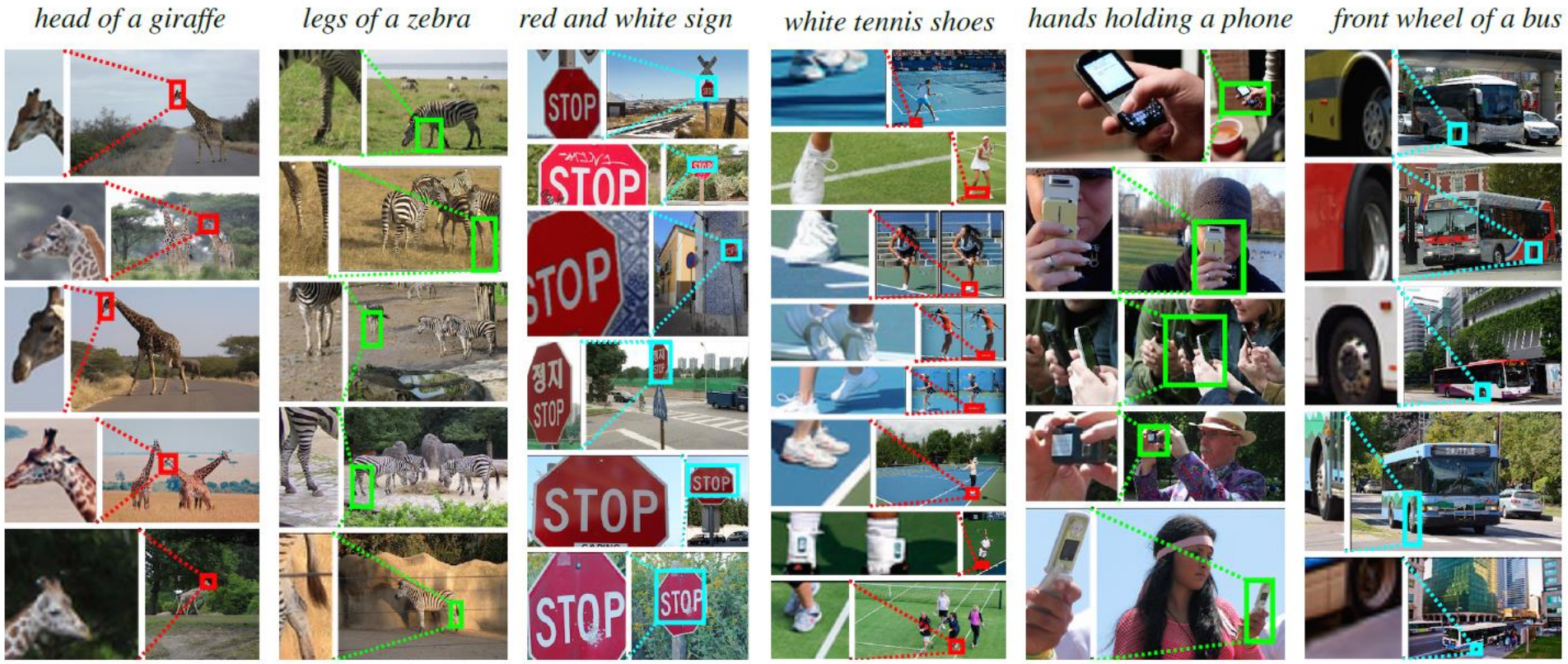
Dense Captioning (Johnson et al.,2015)



Dense Captioning



Towards interpretable image search systems



Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and research directions

Visual Question Answering

How many elephants are there?

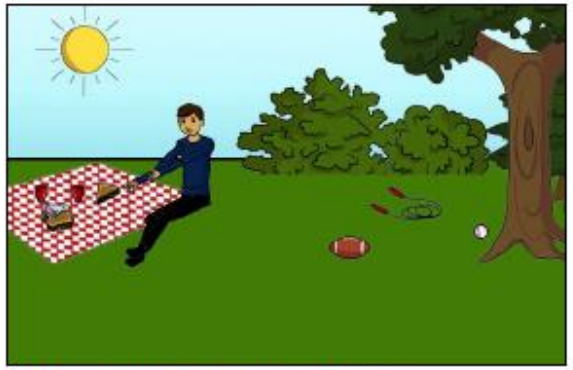
Answer

Answer	Confidence
2	0.4350
3	0.2789
1	0.0766
4	0.0664
5	0.0328

Visual QA dataset (Agrawal et al., 2016)

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

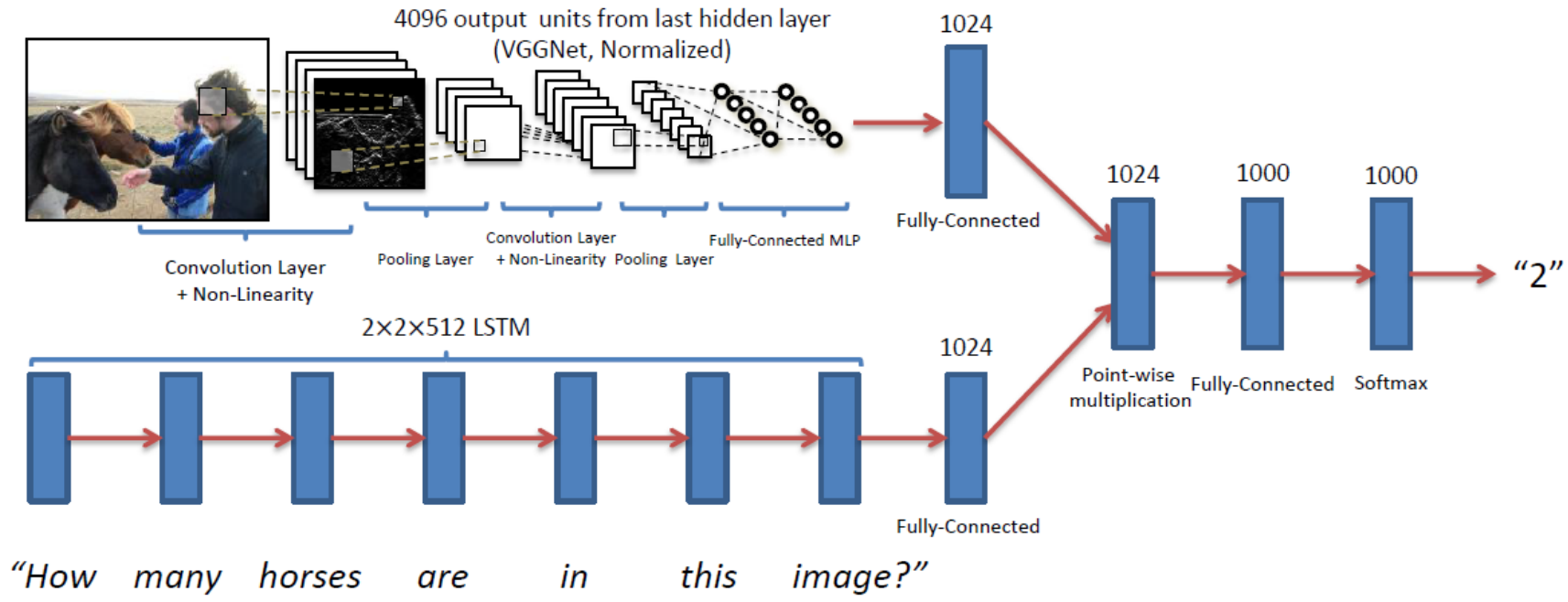
How many slices of pizza are there?
Is this a vegetarian pizza?



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

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

Image/Text Encoder with LSTM answer module (Agrawal et al., 2016)

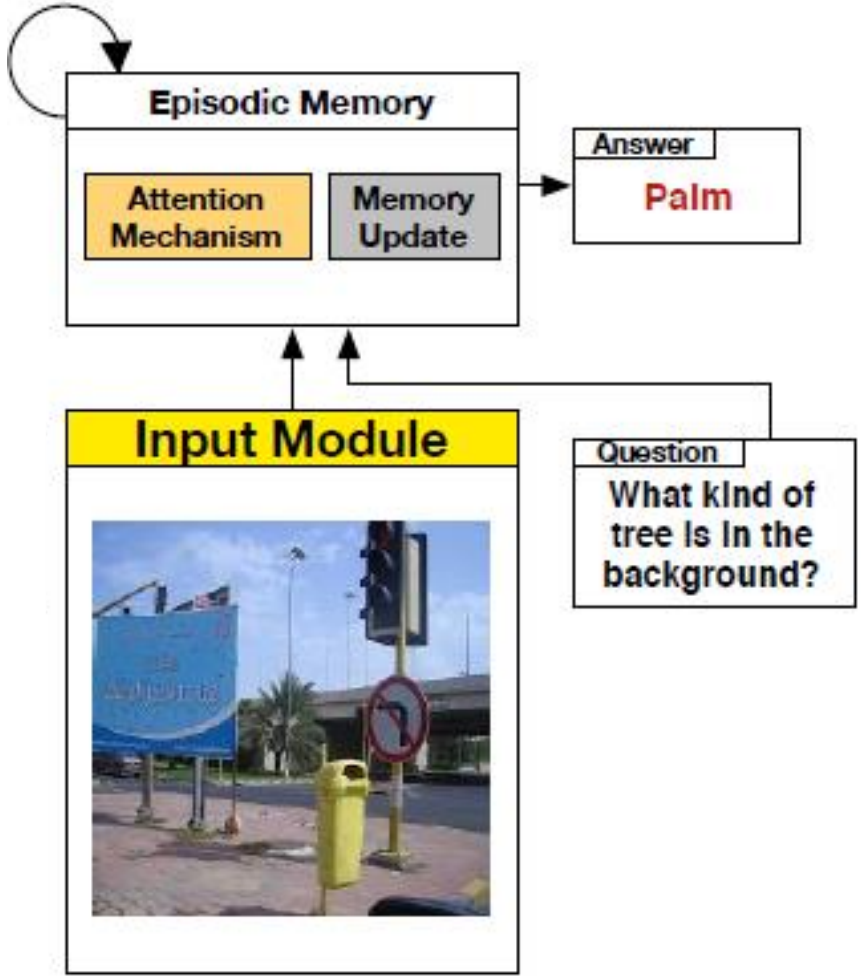


Results

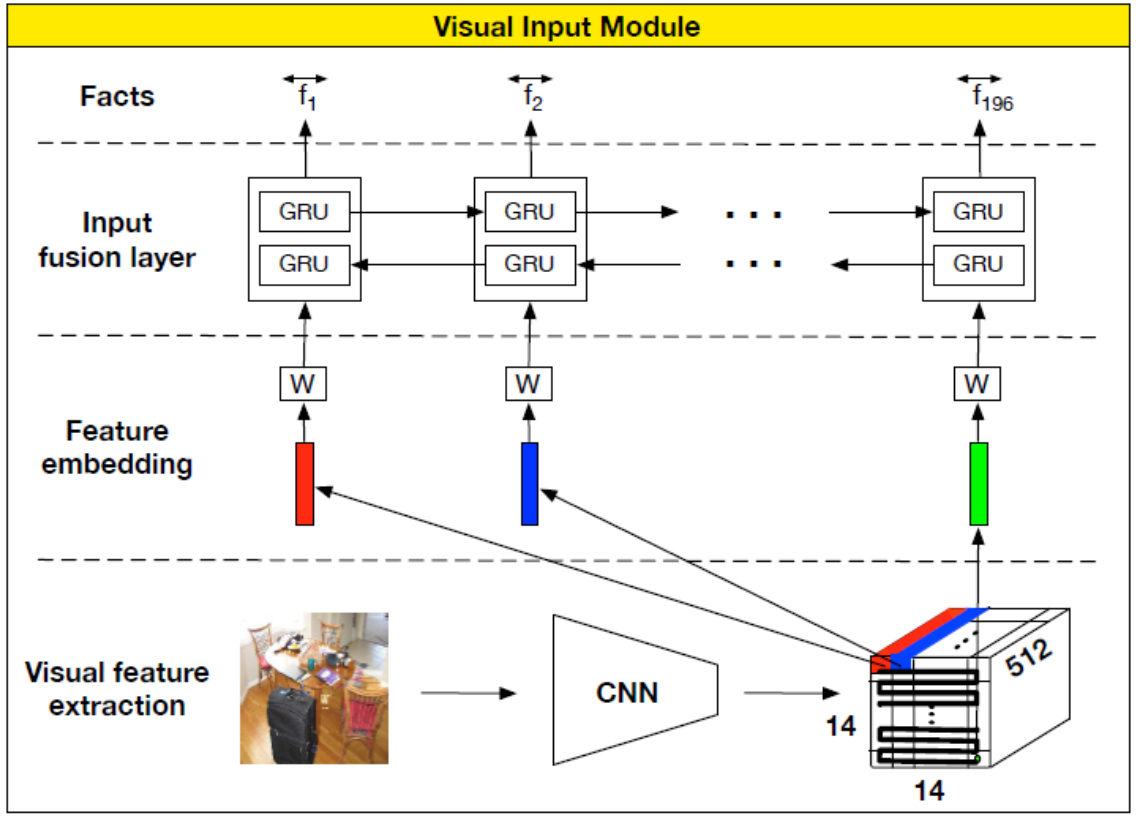
	Open-Ended				Multiple-Choice			
	All	Yes/No	Number	Other	All	Yes/No	Number	Other
prior ("yes")	29.66	70.81	00.39	01.15	29.66	70.81	00.39	01.15
per Q-type prior	37.54	71.03	35.77	09.38	39.45	71.02	35.86	13.34
nearest neighbor	42.70	71.89	24.36	21.94	48.49	71.94	26.00	33.56
BoW Q	48.09	75.66	36.70	27.14	53.68	75.71	37.05	38.64
I	28.13	64.01	00.42	03.77	30.53	69.87	00.45	03.76
BoW Q + I	52.64	75.55	33.67	37.37	58.97	75.59	34.35	50.33
LSTM Q	48.76	78.20	35.68	26.59	54.75	78.22	36.82	38.78
LSTM Q + I	53.74	78.94	35.24	36.42	57.17	78.95	35.80	43.41
deeper LSTM Q	50.39	78.41	34.68	30.03	55.88	78.45	35.91	41.13
deeper LSTM Q + norm I	**57.75**	**80.50**	**36.77**	**43.08**	**62.70**	**80.52**	**38.22**	**53.01**
Caption	26.70	65.50	02.03	03.86	28.29	69.79	02.06	03.82
BoW Q + C	54.70	75.82	40.12	42.56	59.85	75.89	41.16	52.53

TABLE 2: Accuracy of our methods for the open-ended and multiple-choice tasks on the VQA test-dev for real images. Q = Question, I = Image, C = Caption. (Caption and BoW Q + C results are on val). See text for details.

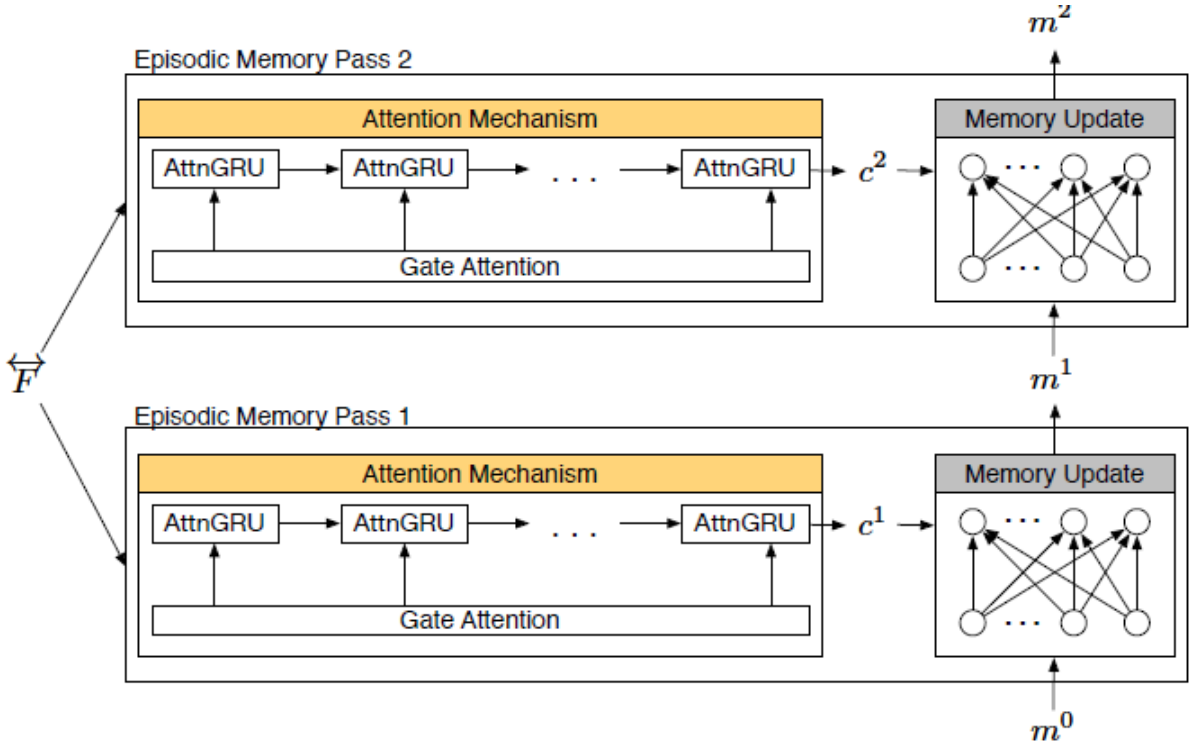
Memory Network based VQA (Xiong et al., 2016)



Visual Input Module



Episodic Memory Module



Results on VQA dataset

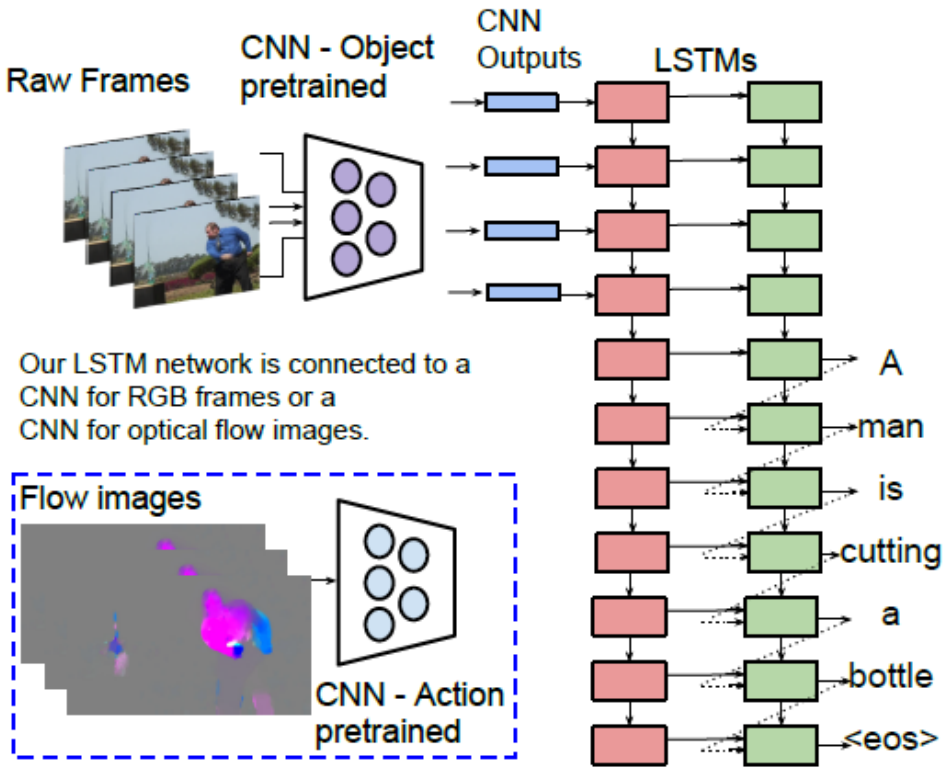
	test-dev				test-std
Method	All	Y/N	Other	Num	All
VQA					
Image	28.1	64.0	3.8	0.4	-
Question	48.1	75.7	27.1	36.7	-
Q+I	52.6	75.6	37.4	33.7	-
LSTM Q+I	53.7	78.9	36.4	35.2	54.1
ACK	55.7	79.2	40.1	36.1	56.0
iBOWIMG	55.7	76.5	42.6	35.0	55.9
DPPnet	57.2	80.7	41.7	37.2	57.4
D-NMN	57.9	80.5	43.1	37.4	58.0
SAN	58.7	79.3	46.1	36.6	58.9
DMN+	**60.3**	80.5	48.3	36.8	**60.4**

Tutorial Outline

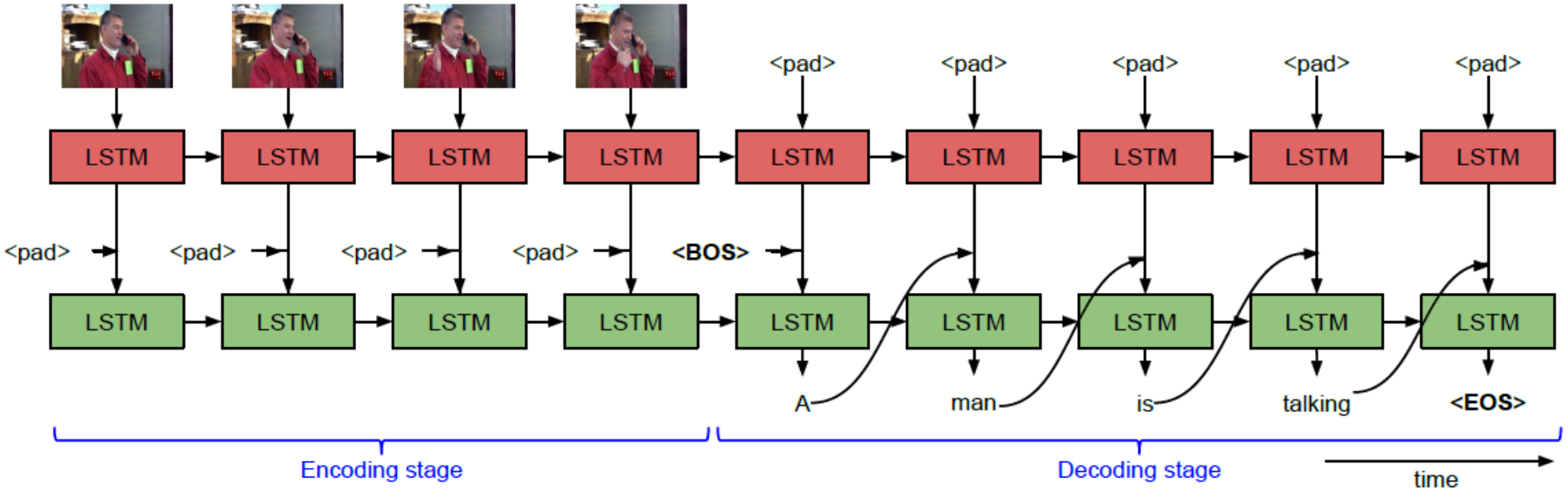
1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and research directions

Video Captioning

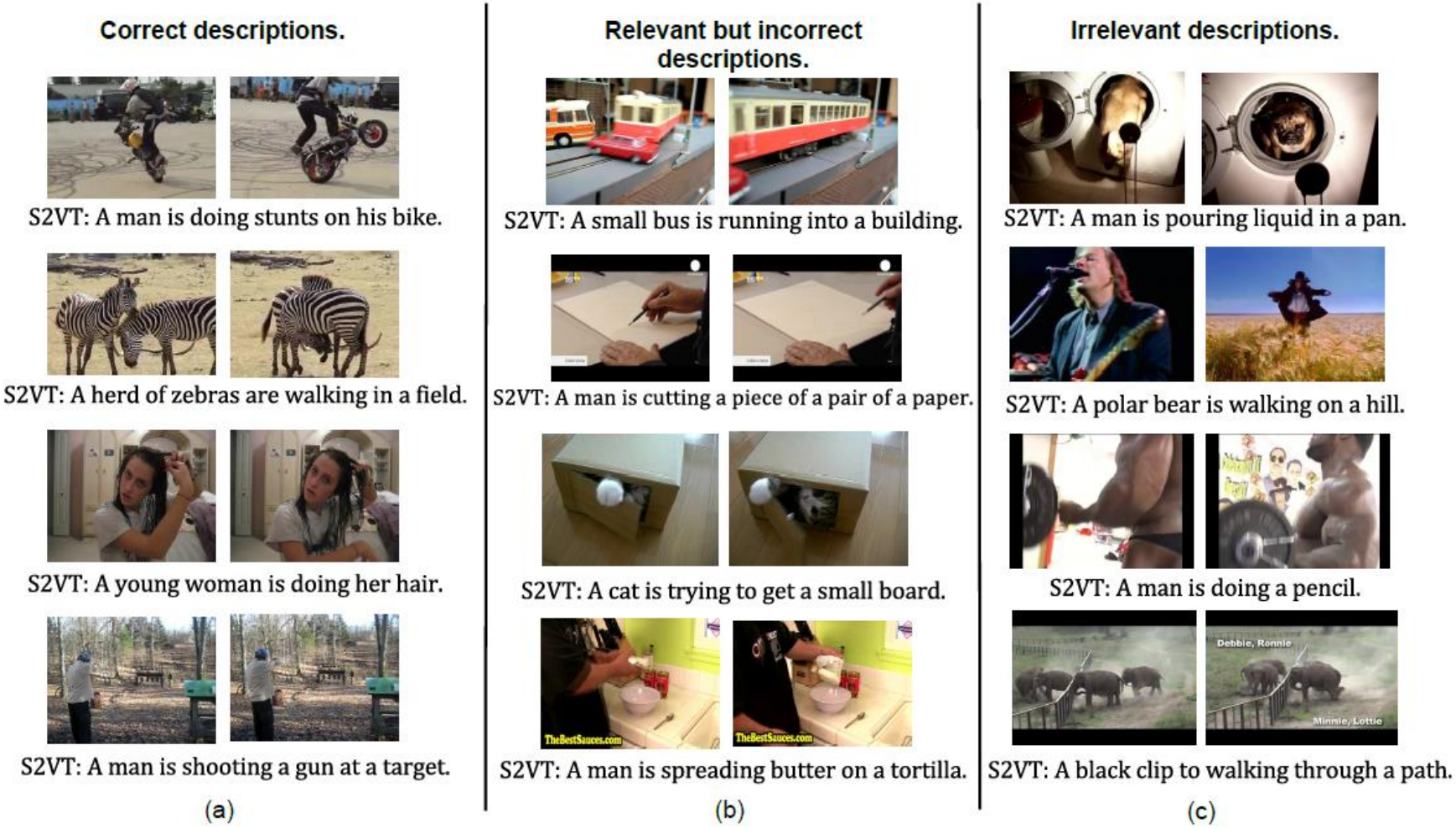
Video Captioning (Venugopalan et al., 2015)



Video Captioning (Venugopalan et al., 2015)



Video Captioning (Venugopalan et al., 2015)



Also..

Describing videos by exploiting temporal structure – (Yao et al., 2015)

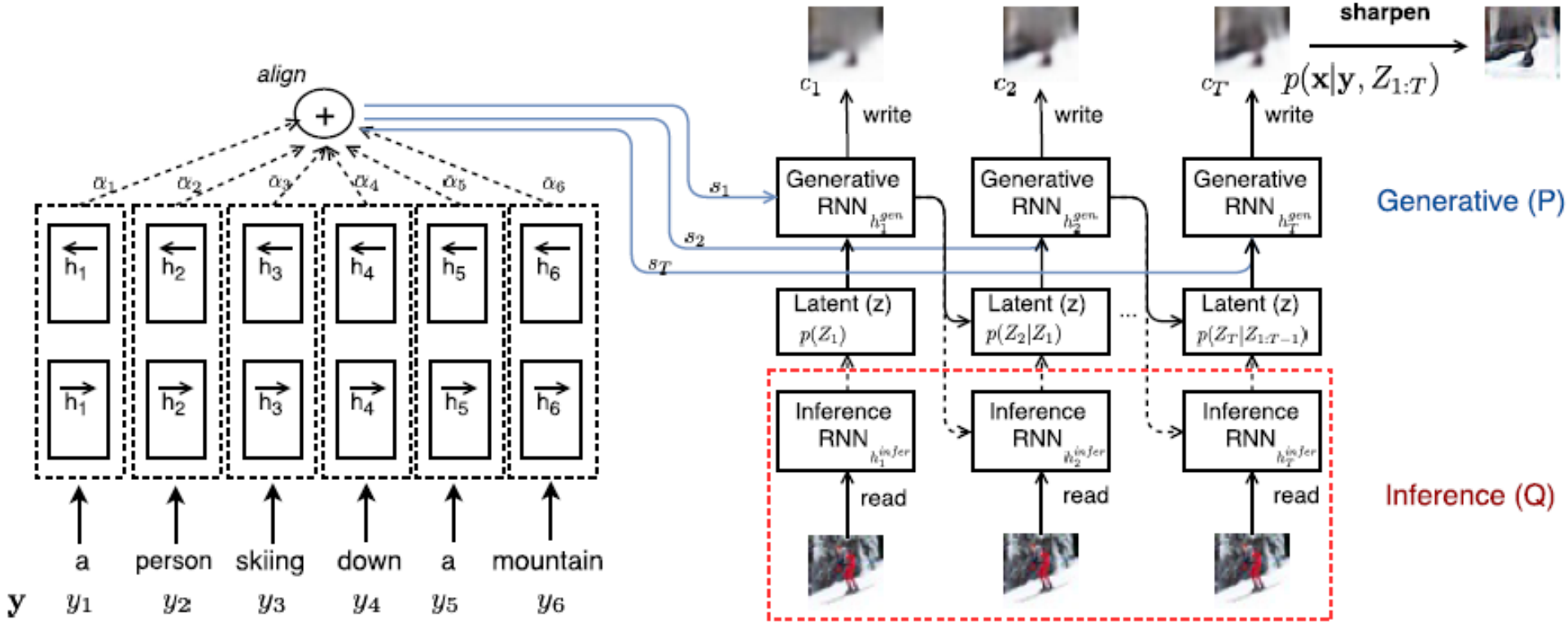
Proposes

- Exploiting local structure by using spatio-temporal convolutional network.
- Exploiting global structure by using temporal attention mechanism.

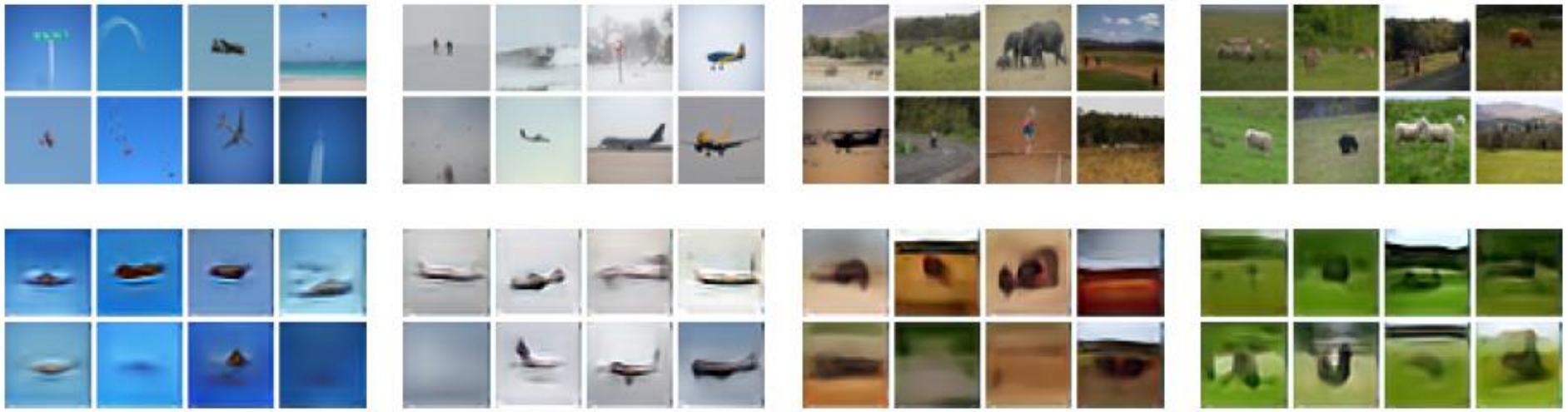
Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
   a) Machine Translation
   b) Image captioning
   c) Visual Question Answering
   d) Video captioning
   e) Image generation from captions
5. Summary and research directions

Generating images from captions (Mansimov et al., 2016)



Generating images from captions (Mansimov et al., 2016)



A very large commercial plane flying in blue skies.

A very large commercial plane flying in rainy skies.

A herd of elephants walking across a dry grass field.

A herd of elephants walking across a green grass field.

Tutorial Outline

1. Introduction and motivation
2. Neural networks – basics
3. Multilingual multimodal representation learning
4. Multilingual multimodal generation
5. Summary and open problems

Summary

- Significant progress in multilingual/multimodal language processing in past few years.
- Reasons
  - Better representation learners
  - Data
  - Compute power
- Finally AI is able to have direct impact in common man's life!
- This is just the beginning!

Research Directions: Representation Learning

- Learn task specific bilingual embeddings

$$maximize \sum_{j \in \{e,f\}} \sum_{i=1}^{T_j} \mathcal{L}(\theta^j) + \lambda \cdot \Omega\left(W^e_{emb}, W^f_{emb}\right) + \mathcal{L}_{task}(\alpha)$$

$w.r.t\ \theta_e, \theta_f, \alpha$

$\theta_e = W_e, W^e_h, W^e_{out}$

$\theta_f = W_f, W^f_h, W^f_{out}$

$\alpha =$ *task –specific parameters*

monolingual similarity *bilingual similarity*

- Learn from comparable corpora (instead of parallel corpora)
- Handle data imbalance
  - More data for $\mathcal{L}(\theta^j)$
  - Less data for $\lambda \cdot \Omega\left(W^e_{emb}, W^f_{emb}\right)$
- Handle larger vocabulary

Research Directions: Language Generation

- Handling large vocabulary during generation
  - hierarchical softmax (Morin & Bengio, 2005)
  - NCE (Mnih & Teh, 2012)
  - Hashing based approaches (Shrivastava & Li, 2014)
  - Sampling based approaches (Jean et al ., 2015)
  - Blackout (Ji et al., 2016)
- Character level instead of word level?
  - Character level decoder - Chung et al., 2016
  - Character level encoder/decoder – Ling et al, 2015
- Still an open problem.

Research Directions: Language Generation

- Handling out-of-vocabulary words.

- Out-of-vocabulary word in source side?
  - Pointing the unknown words - Caglar et al., 2016

- Out-of-vocabulary word in target side?

Research Directions: Language Generation

- Better evaluation metrics?
- Perplexity and BLEU does not encourage diversity in the generation.
- Human evaluation?
- Can we come up with better automated evaluation measures?
- Related work: How NOT to evaluate your dialogue system (Liu et al., 2016)

Research Directions: Language Generation

- Transfer learning for low resource languages?
- Data efficient architectures?

Acknowledgements

Many images/tables are directly taken from the respective papers.

Slides will be made available at:

http://sarathchandar.in/mmnlp-tutorial/

Bibliography of related papers will be maintained at:

http://github.com/apsarath/mmnlp-papers

Questions?