

KING LEAR:

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

Who wrote this play?

- A. William Shakespeare
- B. Ben Jonson
- C. Molière (translation)
- D. Shakespeare's ghostwriter
- E. A machine

[Image credits: Andrej Karpathy (2015) karpathy.github.io
"The Unreasonable Effectiveness of Recurrent Neural Networks"]

Learning representations of text for natural language processing

Piotr Mirowski, Google DeepMind

Table of contents

- ❖ **Representing words**

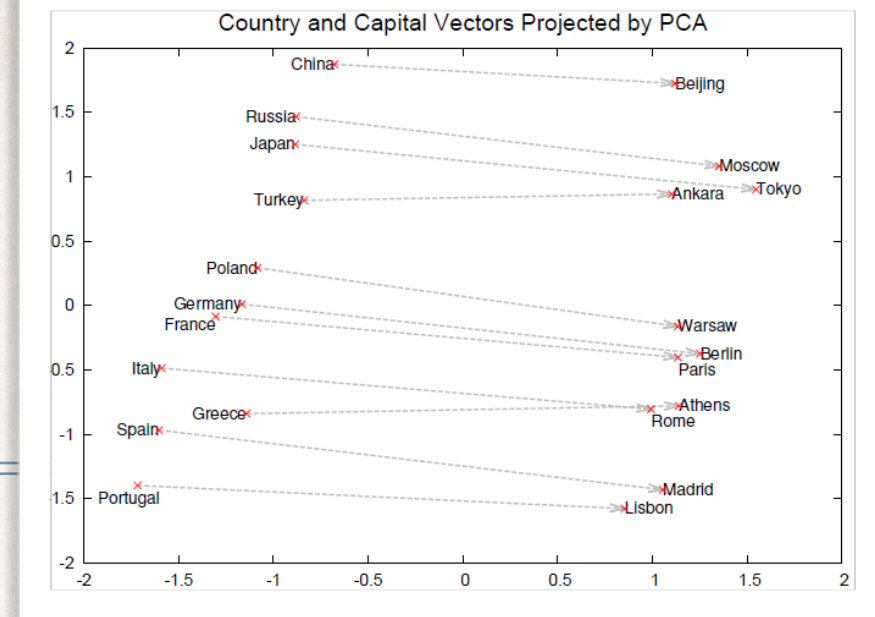
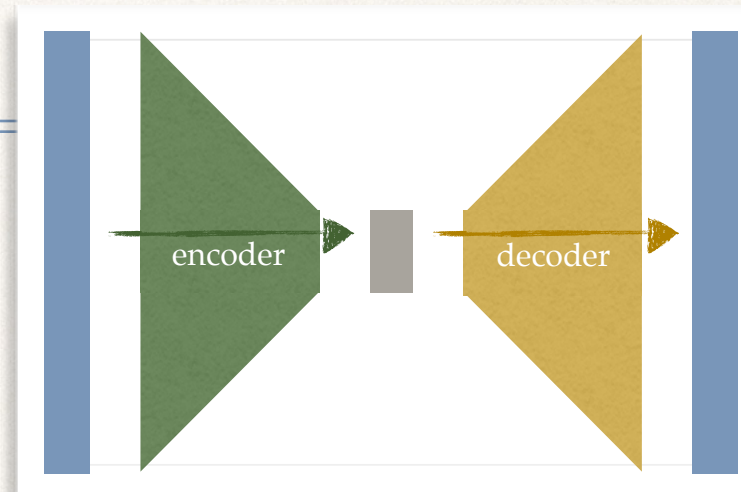
- ❖ Distributional Semantics
- ❖ Skip-grams and **word2vec**
- ❖ Sentence completion

- ❖ **Neural language models**

- ❖ N-grams and language modeling
- ❖ Recurrent Neural Networks (RNNs) and **RNNLM**
- ❖ Speech recognition

- ❖ **Recent developments**

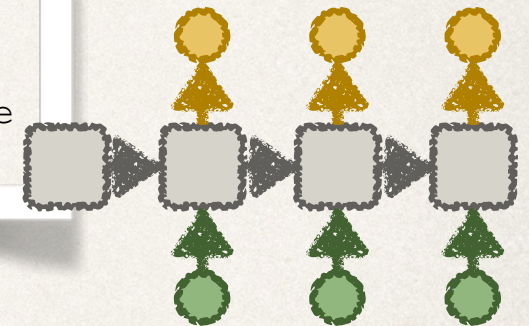
- ❖ Long Short-Term Memory RNNs
- ❖ Sentence-to-sentence machine translation
- ❖ Image captioning



[Image credits: Mikolov et al (2013a)]

cheapest flights from seattle to _

the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...



A person riding a motorcycle on a dirt road.



A group of young people playing a game of frisbee.



A herd of elephants walking across a dry grass field.



[Image credits: Vinyals et al (2014)]

Table of contents

- ❖ **Representing words**

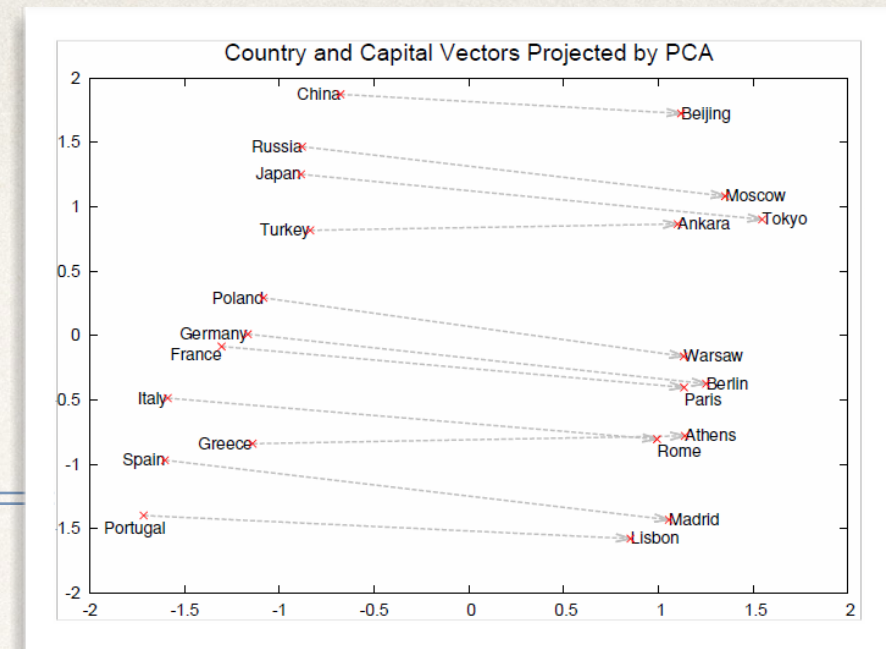
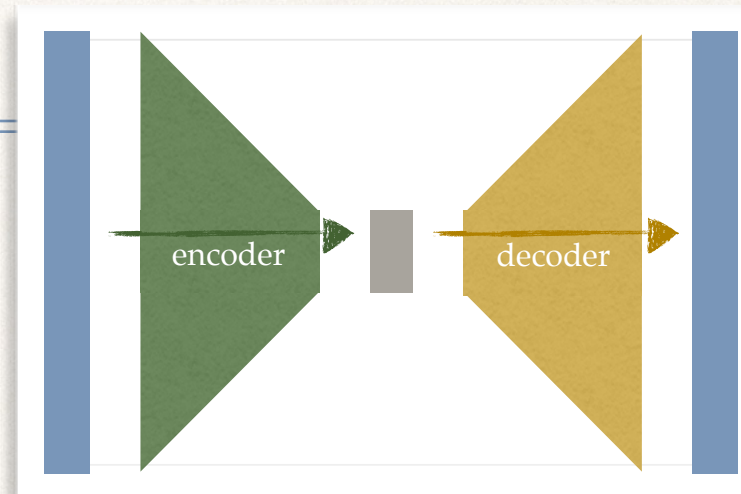
- ❖ Distributional Semantics
- ❖ Skip-grams and **word2vec**
- ❖ Sentence completion

- ❖ Neural language models

- ❖ N-grams and language modeling
- ❖ Recurrent Neural Networks (RNNs) and RNNLM
- ❖ Speech recognition

- ❖ Recent developments

- ❖ Long Short-Term Memory RNNs
- ❖ Sentence-to-sentence machine translation
- ❖ Image captioning



[Image credits: Mikolov et al (2013a)]

cheapest flights from seattle to _

Motivation: choose the word “that makes most sense”

- ❖ Sentence completion
- ❖ Search query formulation
- ❖ Question answering
- ❖ ...

what to cook with broccoli and _

what to cook with broccoli and **beef**

what to cook with broccoli and **butter**

what to cook with broccoli and **blenders**

what to cook with broccoli and **boomboxes**

Distributional semantics

- ❖ Word = vector of “features”

- ❖ How to quantify
semantic similarity
between words

- ❖ **Distributional hypothesis:**
words in similar contexts
have similar meanings

[Zellig Harris (1954) “Distributional structure”, *Word*]

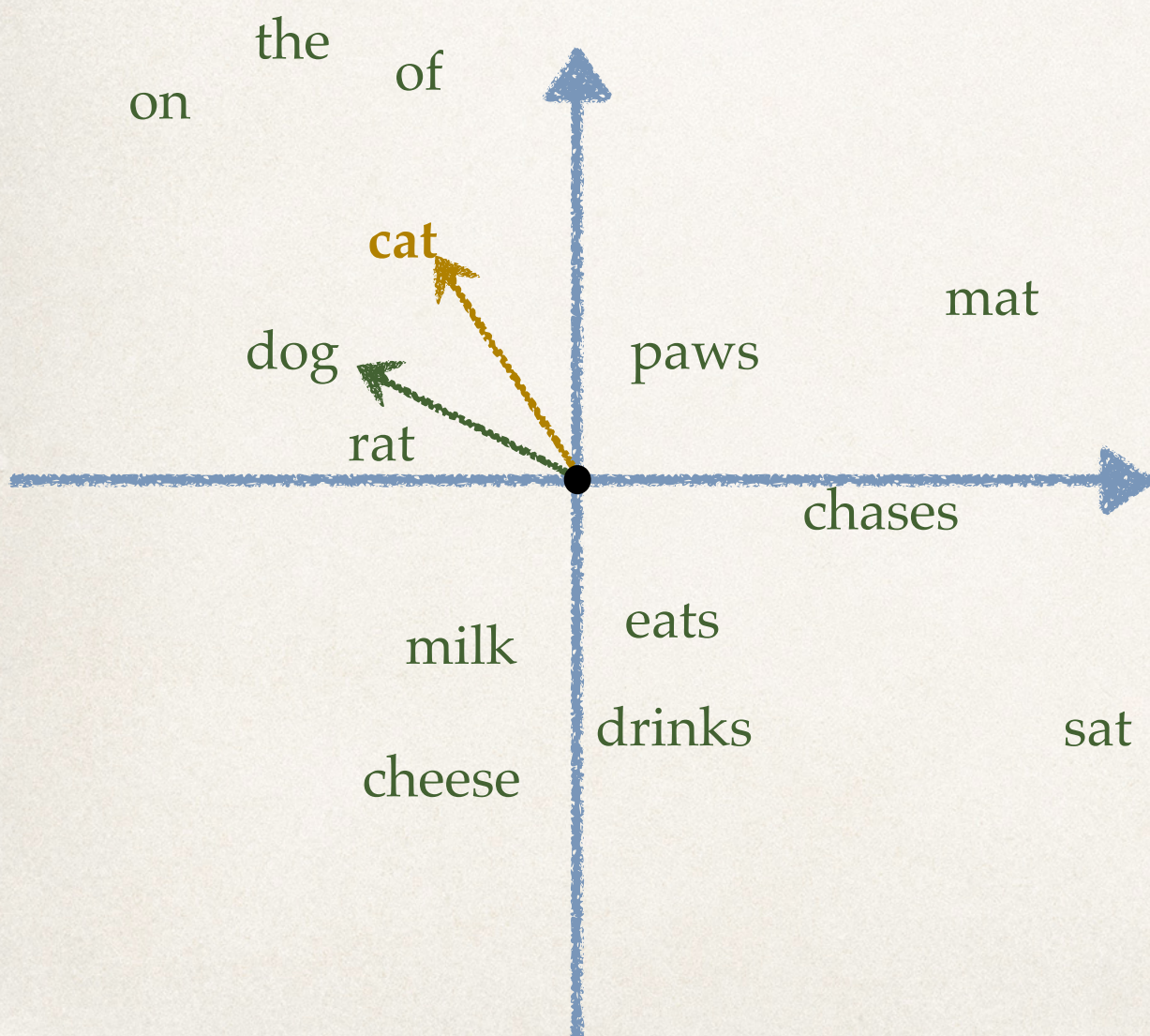
- ❖ “You shall know a word by the company it keeps”

[John R Firth (1957) “Papers in Linguistics 1934-1951”,
London Oxford University Press]

on the of mat
cat
dog paws
rat
chases
milk eats sat
cheese drinks

Semantic similarity

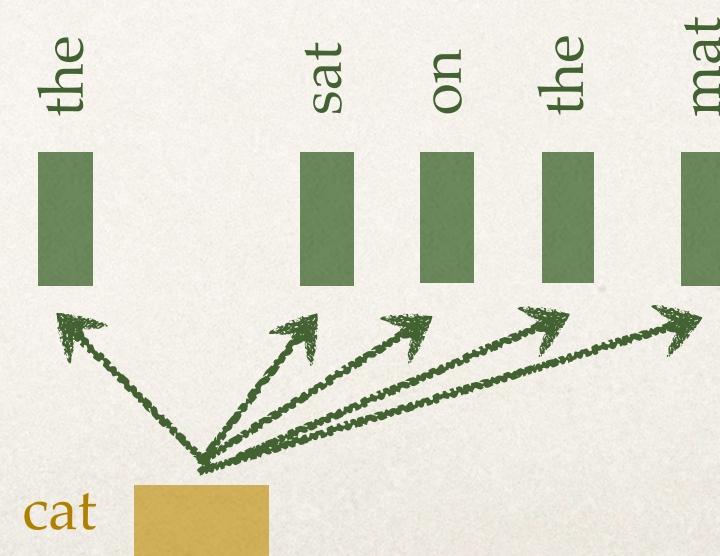
Vector-space representation
of word vectors



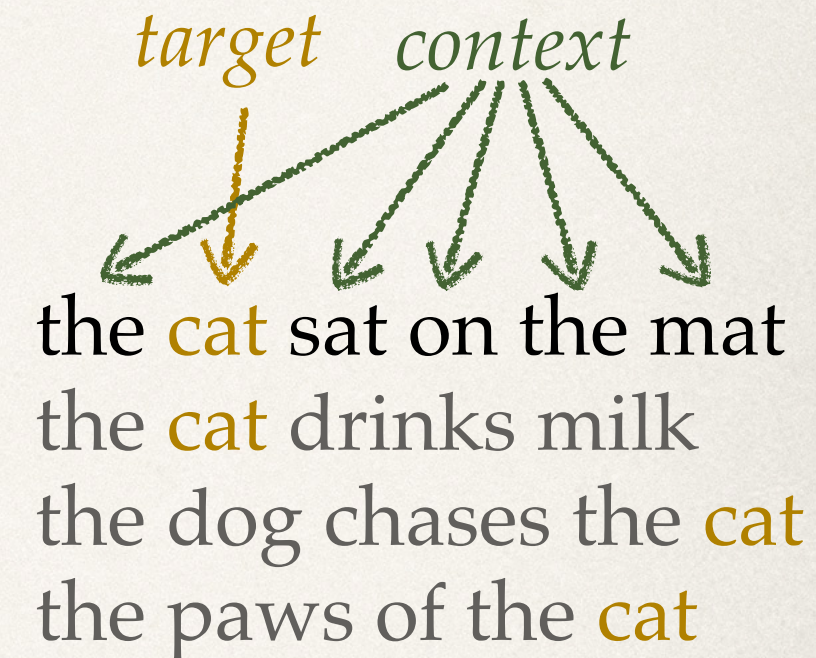
Vector-space cosine similarity
between words w and v

$$\cos(w, v) = \frac{\mathbf{z}_w^T \mathbf{z}_v}{\|\mathbf{z}_w\|_2 \|\mathbf{z}_v\|_2}$$

the **cat** sat on the mat



Word co-occurrence



the cat chases the *rat*
the *rat* eats cheese
the rat eats *the* mat

Word co-occurrence matrix

context

cat chases cheese dog drinks eats mat milk of on paws rat sat the

target

cat

$$m_{w,c} \propto \frac{\#(w, c)}{\#(c)}$$

M

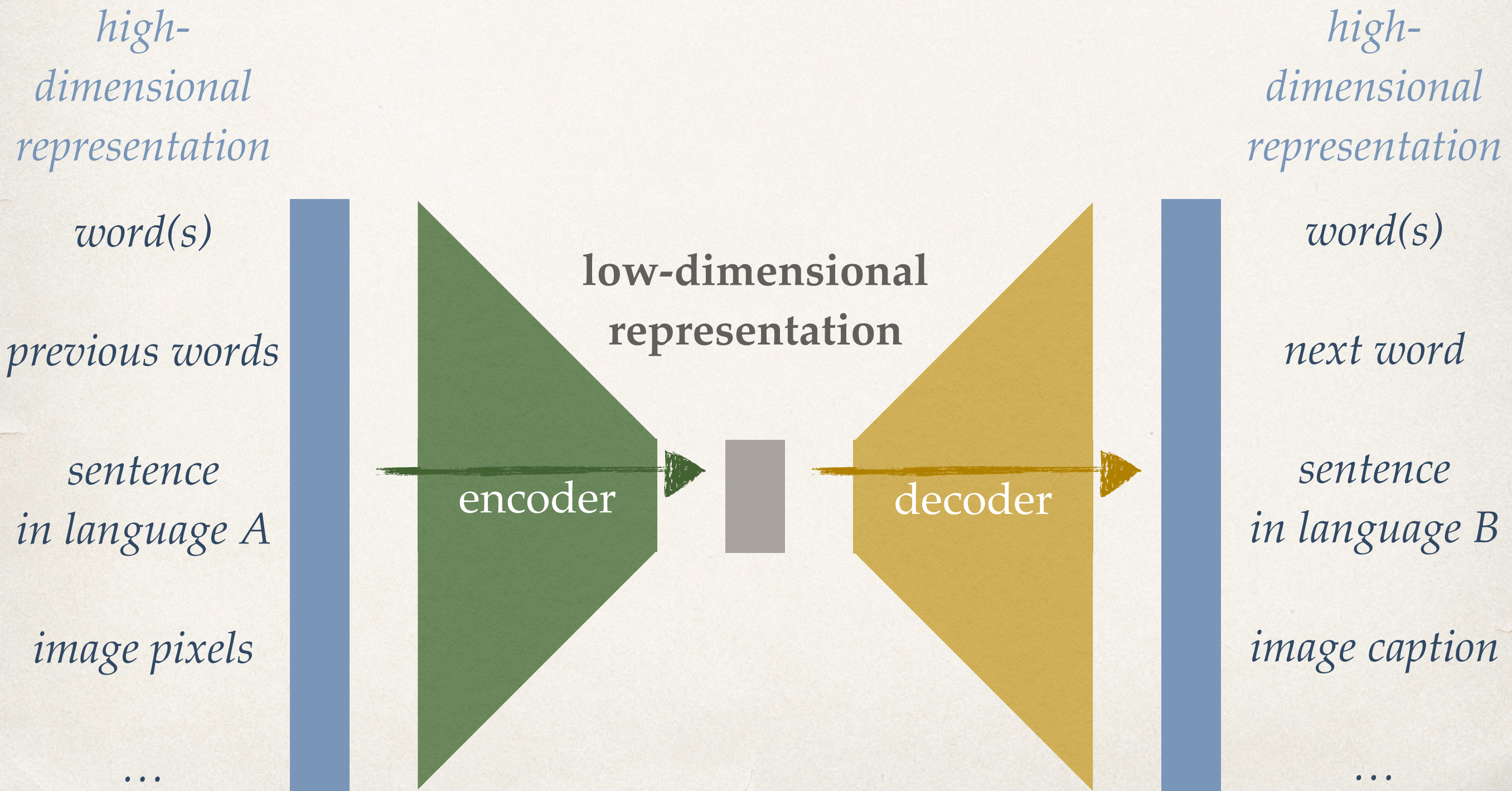
$$m_{w,c} \propto \log \frac{\#(w, c)}{\#(w)\#(c)}$$

rat

the cat sat on the mat
the cat drinks milk
the dog chases the cat
the paws of the cat

the cat chases the rat
the rat eats cheese
the rat eats the mat

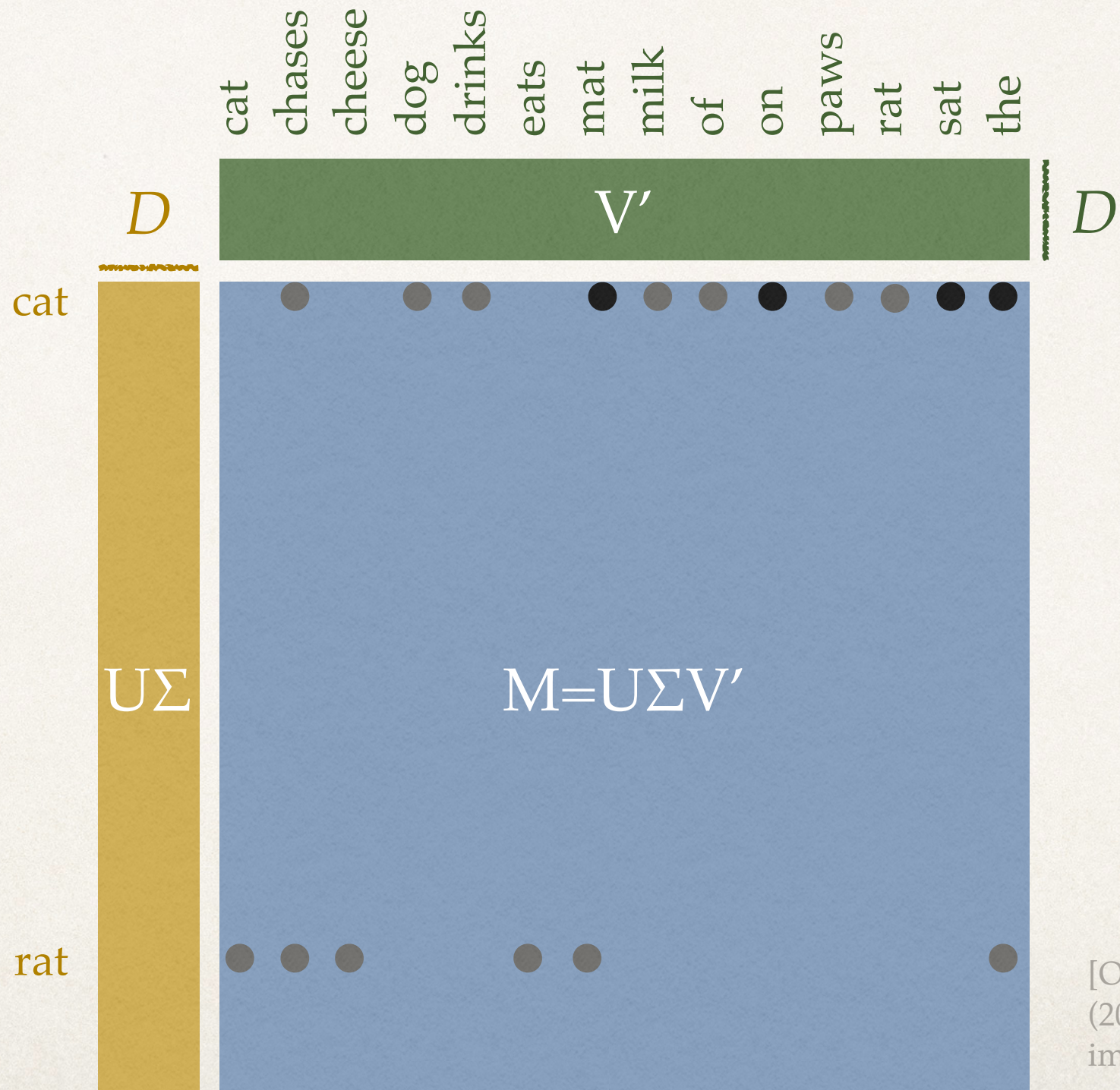
Dimensionality reduction through compression and reconstruction



Representation learning

- ❖ learning distributed representation of symbolic data
[Geoff Hinton (1986)
“Learning distributed representation of concepts”,
Conference of the Cognitive Science Society]
- ❖ “success of machine learning algorithms
generally depends on data representation”
“learn to identify and disentangle the underlying
explanatory factors hidden in the observed [...] data”
[Yoshua Bengio et al. (2014)
“Representation learning: a review and new perspectives”, *arXiv*
1206.5538v3]
- ❖ observed data often lie on **low-dimensional** manifold

Singular Value Decomposition of word co-occurrence matrices

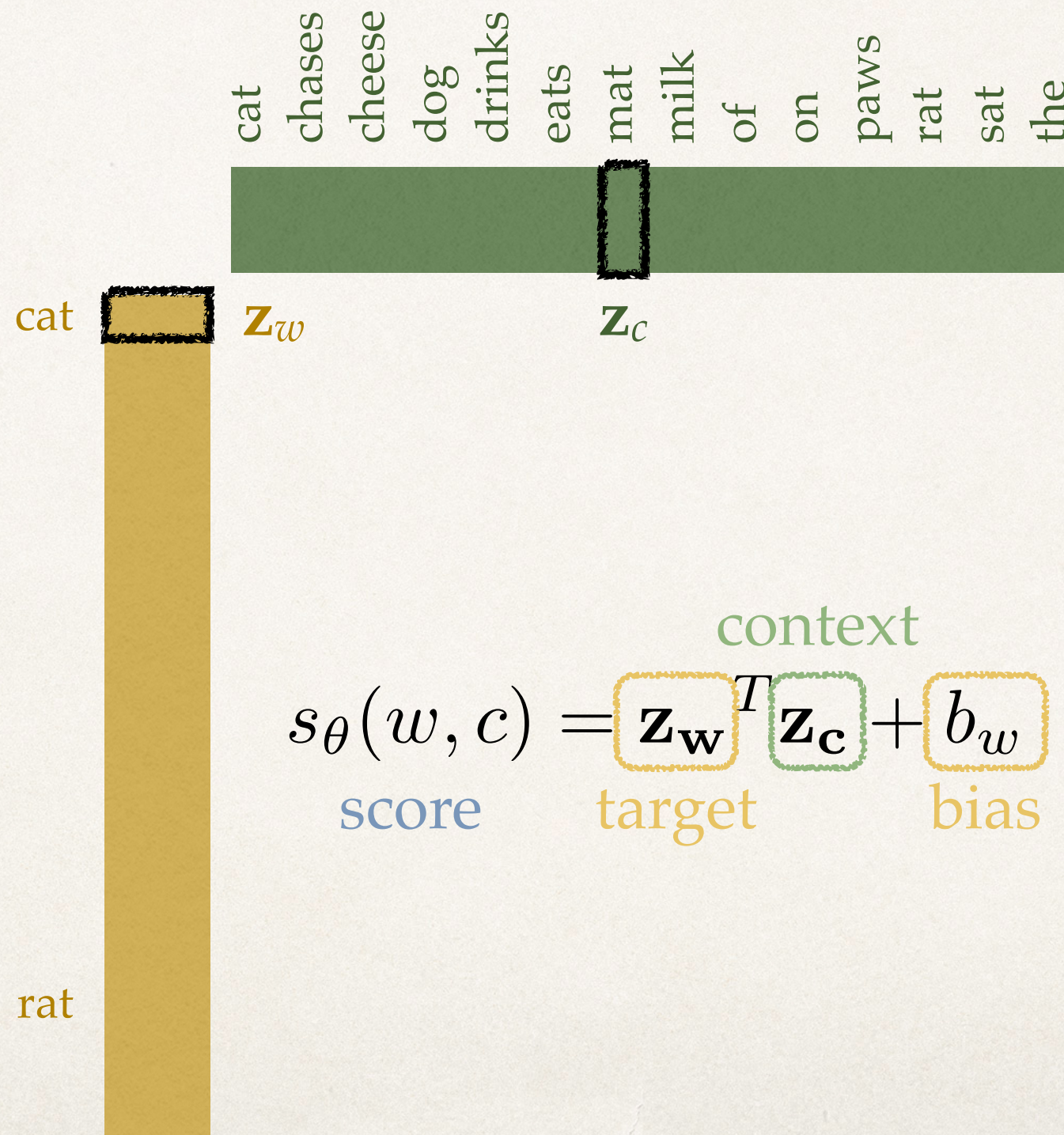


Computationally expensive....

[Omer Levy and Yoav Goldberg (2014) “Neural word embedding as implicit matrix factorization”, *NIPS*]

Word embedding

[Andriy Mnih and Koray Kavukcuoglu (2013) "Learning word embeddings efficiently with noise-contrastive estimation", *NIPS*;
Tomas Mikolov et al. (2013a) "Efficient Estimation of Word Representation in Vector Space", *arXiv* 1301.3781v3;
Tomas Mikolov et al. (2013b) "Distributed Representation of Words and Phrases and Their Compositionality", *NIPS*]



$$s_{\theta}(w, c) = \mathbf{z}_w^T \mathbf{z}_c + b_w$$

Word embeddings

Examples of word embeddings
obtained using word2vec
(code.google.com/p/word2vec)
on 3.2B word Wikipedia,
with 2M-word vocabulary
and word vector dimension $D=200$

debt	decrease	met	slow	france	xbox
debts	increase	meeting	slower	marseille	playstation
repayments	increases	meet	fast	french	wii
repayment	decreased	meets	slowing	nantes	xbla
monetary	greatly	had	slows	vichy	wiiware
payments	decreasing	welcomed	slowed	paris	gamecube

Context-dependent word probability

$$P(w|c) \approx \frac{\#(w, c)}{\#(c)}$$

Learn model (e.g., word embeddings)
parameterized by θ , so that:

“softmax”

$$P(w|c) = \frac{e^{s_{\theta}(w,c)}}{\sum_{v=1}^V e^{s_{\theta}(v,c)}}$$

correct answer

normalization term

Maximum likelihood learning

Learn model (e.g., word embeddings)
parameterized by θ , so that:

“softmax”

$$P(w|c) = \frac{e^{s_{\theta}(w,c)}}{\sum_{v=1}^V e^{s_{\theta}(v,c)}}$$

correct answer

normalization term

maximize

$$\log P(w|c) = s_{\theta}(w,c) - \log \sum_{v=1}^V e^{s_{\theta}(v,c)}$$

correct answer

normalization term

Maximum likelihood learning

Stochastic gradient ascent (or descent):
after showing each pair (word w , context c),
update the parameters θ

$$\theta \leftarrow \theta + \eta \frac{\partial L(w, c; \theta)}{\partial \theta}$$

$$L(w, c; \theta) = \log P(w|c) = \underbrace{s_{\theta}(w, c)}_{\text{correct answer}} - \underbrace{\log \sum_{v=1}^V e^{s_{\theta}(v, c)}}_{\text{normalization term}}$$

High computational complexity of learning word embeddings

high-dimensional
normalization term
(e.g., $V > 100k$ words)

$$P(w|c) = \frac{e^{s_{\theta}(w,c)}}{\sum_{v=1}^V e^{s_{\theta}(v,c)}}$$

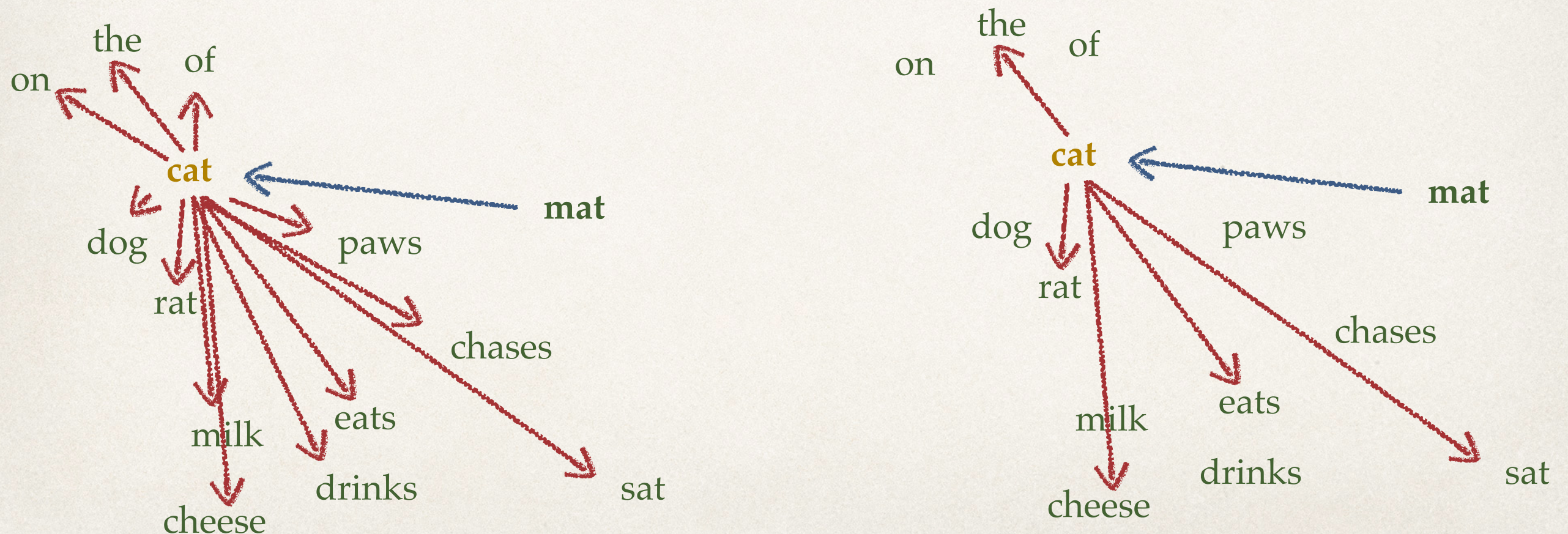
Solution #1:
parallelize
on a GPU

Solution #2:
approximate
normalization term

Approximate maximum likelihood word embeddings: skip-grams

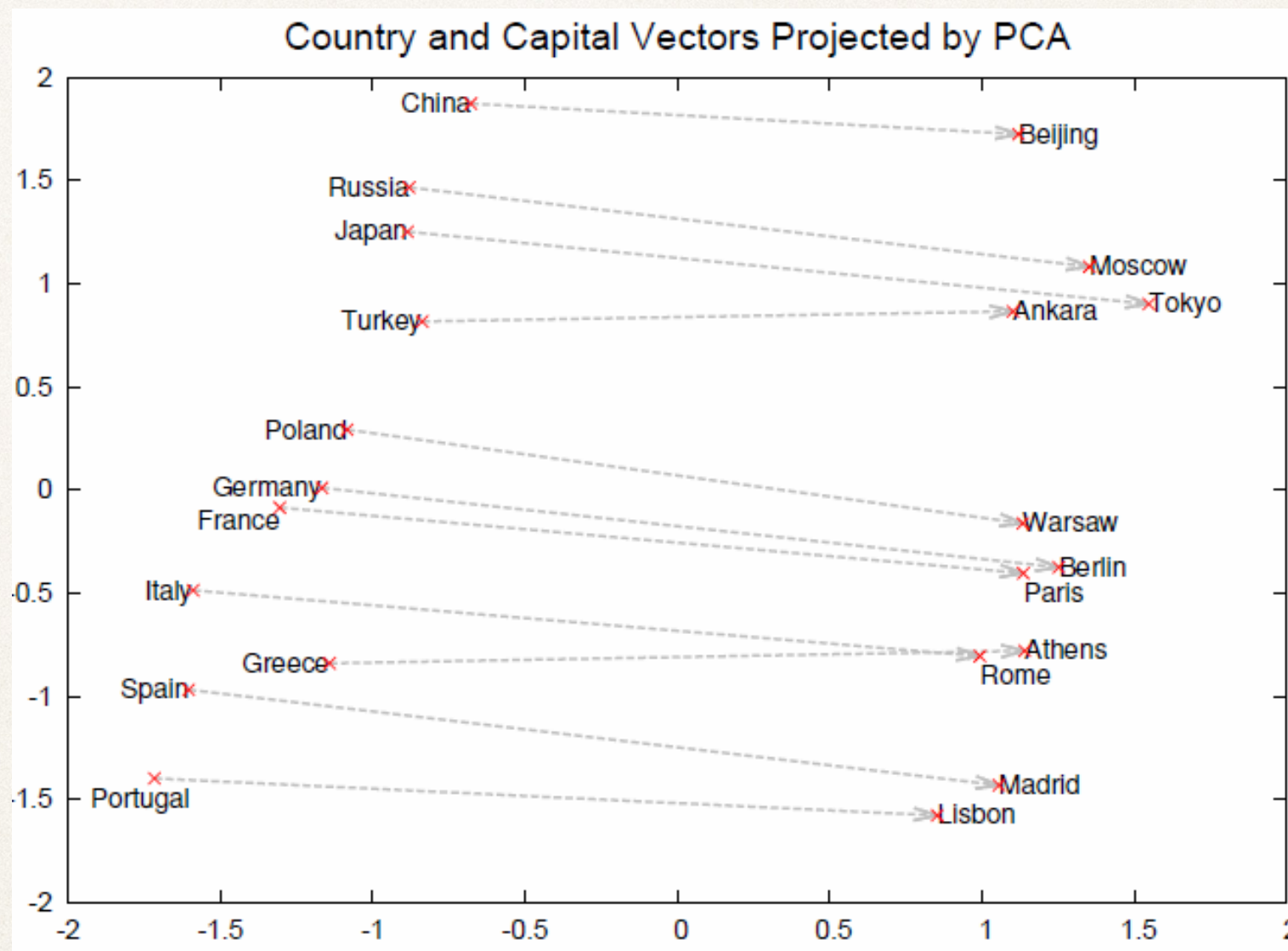
Replace **normalization term** on entire vocabulary
For each correct answer, select **only K wrong answers**
(sampled based on their frequency)

the **cat** sat on the **mat**



Semantic analogy task:

king - man + woman = ? (queen)



[Image credits: Mikolov et al (2013a)]

[Mnih & Kavukcuoglu, 2013;
Mikolov et al, 2013a, 2013b;
Levy & Goldberg, 2014]

Sentence completion task

- ❖ Microsoft Research Sentence Completion Task

[Geoff Zweig and Chris Burges (2011), “The Microsoft Research Sentence Completion Challenge”, *MSR Technical Report*]

- ❖ Training set:

- ❖ ~520 novels (19th century)

- ❖ 48M words

- ❖ Evaluation on 1024 sentences

- ❖ From 5 Sherlock Holmes novels

- ❖ 1 missing word, 5 choices:

- ❖ 1 ground truth

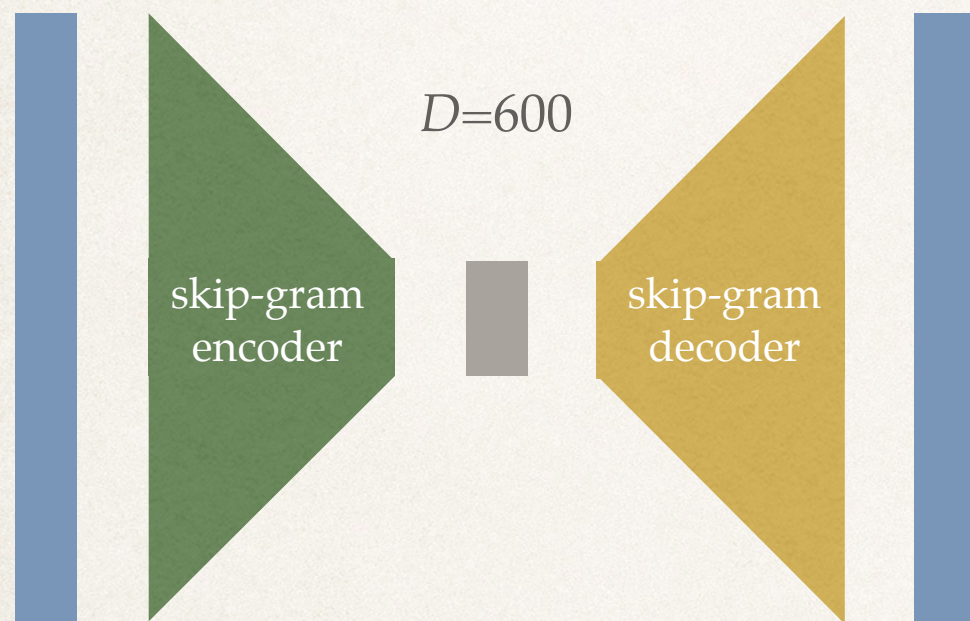
- ❖ 4 grammatically correct impostors

That is his	generous	fault, but on the whole he's a good worker.
That is his	mother's	fault, but on the whole he's a good worker.
That is his	successful	fault, but on the whole he's a good worker.
That is his	main	fault, but on the whole he's a good worker.
That is his	favourite	fault, but on the whole he's a good worker.

Sentence completion task

context
word

missing
word



Algorithm	Accuracy (test set)
random	20%
SVD (word-paragraph)	49%
skip-gram	48%
smoothed 4-gram	39%
human	90%

[Mikolov et al. (2013a)]


That is his **generous** fault, but on the whole he's a good worker.
That is his **mother's** fault, but on the whole he's a good worker.
That is his **successful** fault, but on the whole he's a good worker.
That is his **main** fault, but on the whole he's a good worker.
That is his **favourite** fault, but on the whole he's a good worker.

#####

Search query completion task

- ❖ AOL search log data (2006)
 - ❖ 1M distinct queries
 - ❖ Context: partially-formulated *query prefix*
 - ❖ $V=100k$ *query suffixes*
 - ❖ re-rank query completions based on features that include **cosine similarity** of embeddings of query *prefix* and *suffix*

cheapest flights from seattle to	
cheapest flights from seattle to	<i>dc</i>
cheapest flights from seattle to	<i>washington dc</i>
cheapest flights from seattle to	<i>bermuda</i>
cheapest flights from seattle to	<i>bahamas</i>
cheapest flights from seattle to	<i>aruba</i>
cheapest flights from seattle to	<i>punta cana</i>
cheapest flights from seattle to	<i>airport</i>
cheapest flights from seattle to	<i>miami</i>



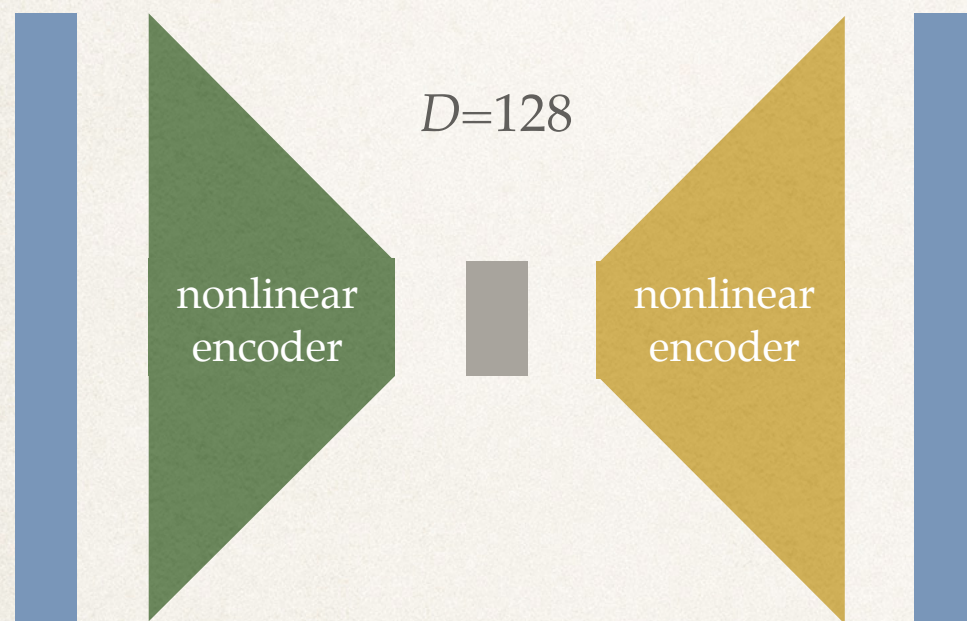
what to cook with chicken and broccoli and	
what to cook with chicken and broccoli and	<i>bacon</i>
what to cook with chicken and broccoli and	<i>noodles</i>
what to cook with chicken and broccoli and	<i>brown sugar</i>
what to cook with chicken and broccoli and	<i>garlic</i>
what to cook with chicken and broccoli and	<i>orange juice</i>
what to cook with chicken and broccoli and	<i>beans</i>
what to cook with chicken and broccoli and	<i>onions</i>
what to cook with chicken and broccoli and	<i>ham soup</i>

[Bhaskar Mitra and Nick Craswell (2015)
“Query Auto-Completion for Rare Prefixes”, CIKM]

Search query completion task

unfinished
query prefix

query
completion
suffix
 $V=100k$



cheapest flights from seattle to	
cheapest flights from seattle to	<i>dc</i>
cheapest flights from seattle to	<i>washington dc</i>
cheapest flights from seattle to	<i>bermuda</i>
cheapest flights from seattle to	<i>bahamas</i>
cheapest flights from seattle to	<i>aruba</i>
cheapest flights from seattle to	<i>punta cana</i>
cheapest flights from seattle to	<i>airport</i>
cheapest flights from seattle to	<i>miami</i>

what to cook with chicken and broccoli and	
what to cook with chicken and broccoli and	<i>bacon</i>
what to cook with chicken and broccoli and	<i>noodles</i>
what to cook with chicken and broccoli and	<i>brown sugar</i>
what to cook with chicken and broccoli and	<i>garlic</i>
what to cook with chicken and broccoli and	<i>orange juice</i>
what to cook with chicken and broccoli and	<i>beans</i>
what to cook with chicken and broccoli and	<i>onions</i>
what to cook with chicken and broccoli and	<i>ham soup</i>

Table of contents

- ❖ Representing words

- ❖ Distributional Semantics
- ❖ Skip-grams and word2vec
- ❖ Sentence completion

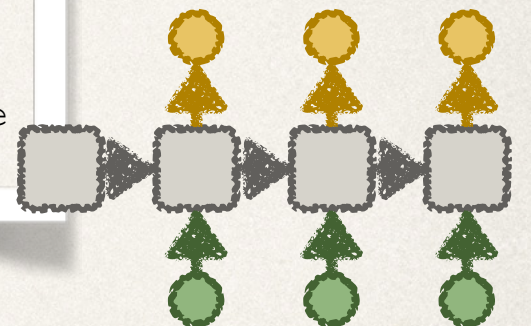
- ❖ **Neural language models**

- ❖ N-grams and language modeling
- ❖ Recurrent Neural Networks (RNNs) and **RNNLM**
- ❖ Speech recognition

- ❖ Recent developments

- ❖ Long Short-Term Memory RNNs
- ❖ Sentence-to-sentence machine translation
- ❖ Image captioning

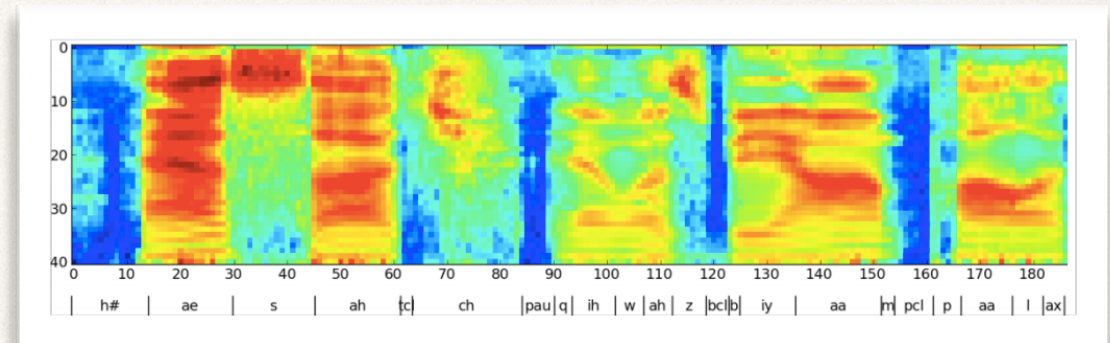
the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...



Motivation: choose the sentence “that makes most sense”

- ❖ Speech recognition
- ❖ Machine translation
- ❖ Sentence completion

Starting from acoustic vectors...



[Graves et al. (2013b) “Speech recognition with deep recurrent neural networks”, *ICASSP*]

... choose the “most likely” sentence

```
the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...
```


N-grams and conditional probabilities of words

<i>context</i>				<i>target</i>	$P(w_t w_{t-1}, w_{t-2}, \dots, w_{t-5})$
the	cat	sat	on	the mat	0.15
w_{t-5}	w_{t-4}	w_{t-3}	w_{t-2}	w_{t-1} w_t	
the	cat	sat	on	the rug	0.12
the	cat	sat	on	the hat	0.09
the	cat	sat	on	the dog	0.01
the	cat	sat	on	the the	0
the	cat	sat	on	the sat	0
the	cat	sat	on	the robot	?
the	cat	sat	on	the printer	?

Chain rule of probability

$$P(w_1, w_2, \dots, w_{T-1}, w_T) = \prod_{t=1}^T P(w_t | w_{t-1}, w_{t-2}, \dots, w_1)$$

the	cat	sat	on	the	mat	$P(w_1)$
the	cat	sat	on	the	mat	$P(w_2 w_1)$
the	cat	sat	on	the	mat	$P(w_3 w_2, w_1)$
the	cat	sat	on	the	mat	$P(w_4 w_3, w_2, w_1)$
the	cat	sat	on	the	mat	$P(w_5 w_4, w_3, w_2, w_1)$
the	cat	sat	on	the	mat	$P(w_6 w_5, w_4, w_3, w_2, w_1)$

Chain rule of probability and n-gram approximation

$$P(w_1, w_2, \dots, w_{T-1}, w_T) \approx \prod_{t=1}^T P(w_t | w_{t-1}, \dots, w_{t-n+1})$$

the	cat	sat	on	the	mat	$P(w_1)$
the	cat	sat	on	the	mat	$P(w_2 w_1)$
the	cat	sat	on	the	mat	$P(w_3 w_2, w_1)$
the	cat	sat	on	the	mat	$P(w_4 w_3, w_2)$
the	cat	sat	on	the	mat	$P(w_5 w_4, w_3)$
the	cat	sat	on	the	mat	$P(w_6 w_5, w_4)$

Language models

- ❖ Quantify likelihood of text

$$P(w_1, w_2, \dots, w_{T-1}, w_T) \approx \prod_{t=1}^T P(w_t | w_{t-1}, \dots, w_{t-n+1})$$

- ❖ Goal: rank n-best lists

- ❖ Machine translation

- ❖ Speech recognition

e.g., re-rank 100-best list returned
by acoustic model and choose sentence
with highest combined
language model and
acoustic model log-likelihood (score)

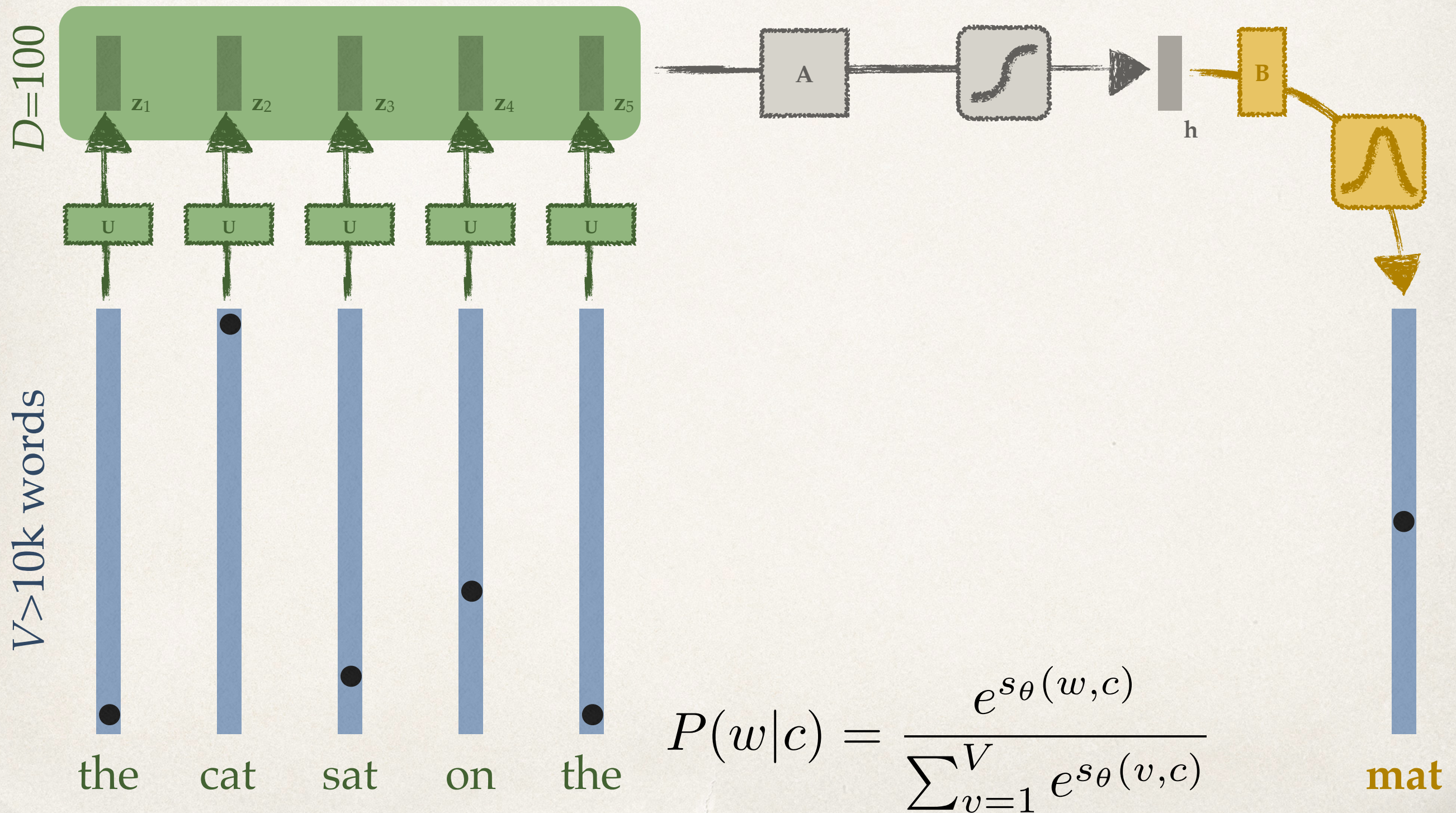
```
the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...
```


Limitations of n-grams

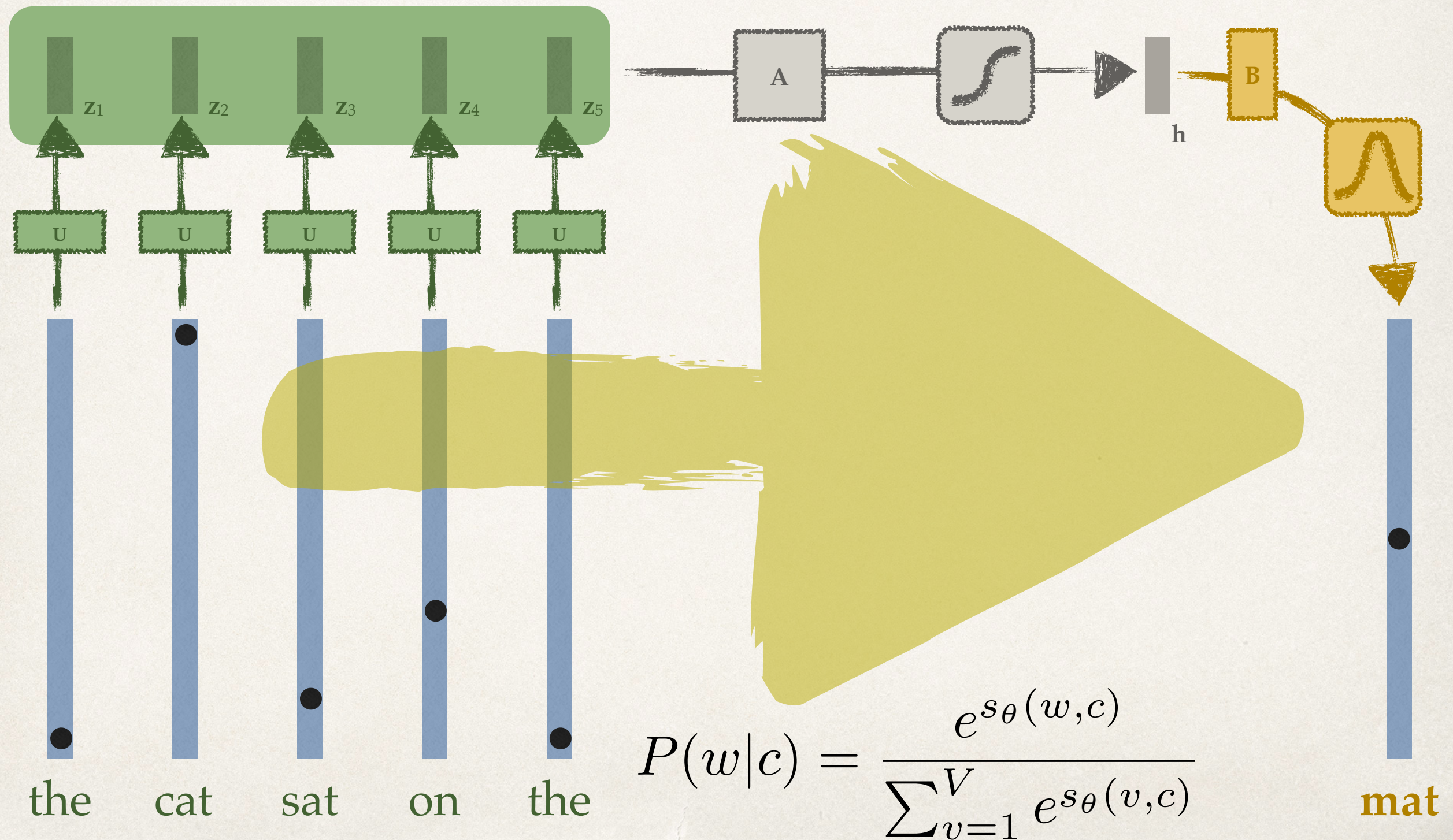
- ❖ *Curse of dimensionality:*
 n -grams need exponential number of examples for a vocabulary of V words:
 V^n possible n-grams
- ❖ No notion of word similarity...
... motivates use of **word embeddings**
~ “**high-dimensional interpolation**”

Neural Probabilistic Language Models

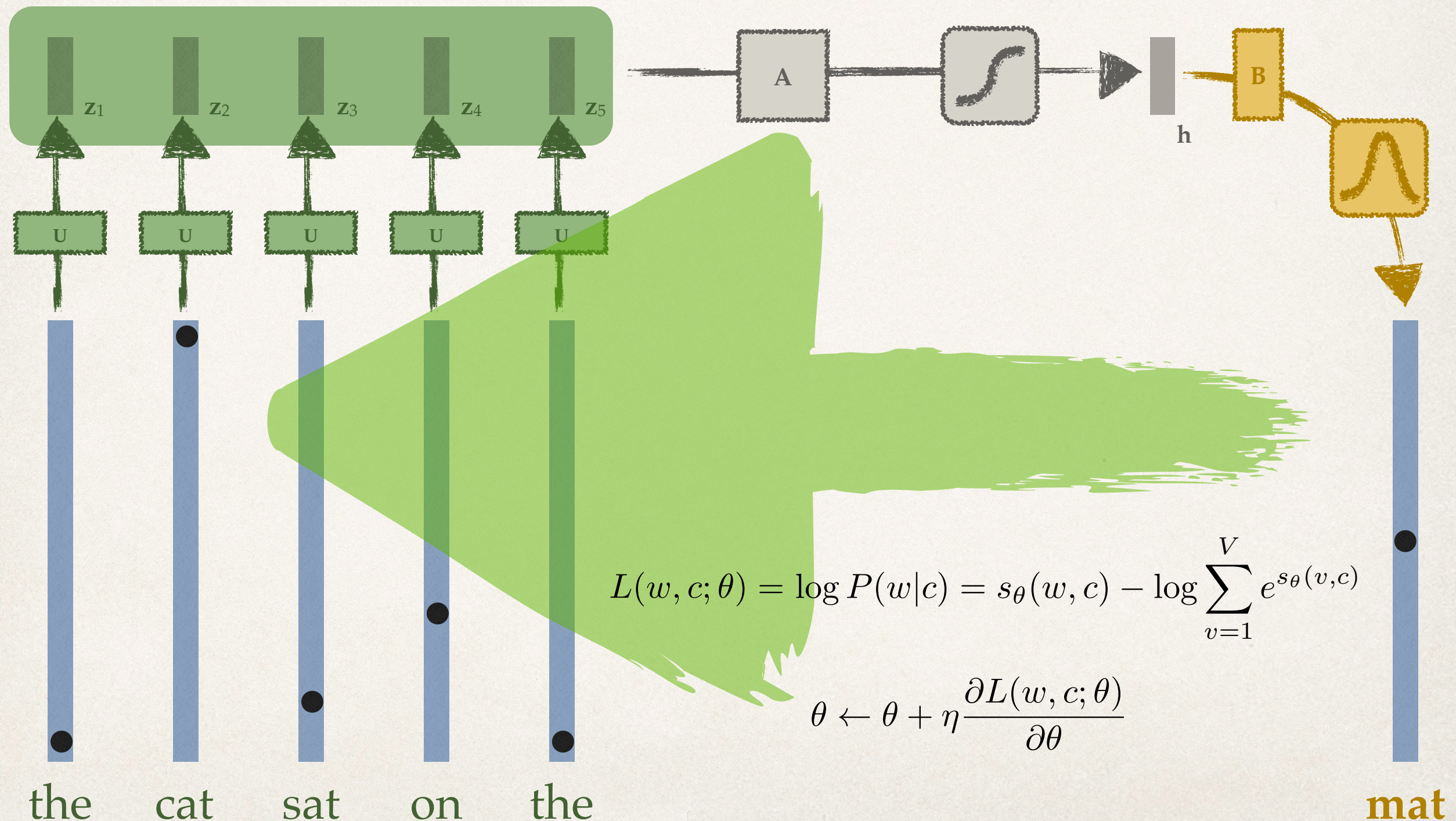
[Yoshua Bengio et al. (2001, 2003), "A Neural Probabilistic Language Model", *JMLR*;
Andriy Mnih and Geoff Hinton, "Three new graphical models for statistical language modeling", *ICML*]



Learning neural language models: step 1: forward propagation



Learning neural language models: step 2: gradient back-propagation



Speech recognition

[Holger Schwenk and Jean-Luc Gauvain (2002)
“Connectionist language modeling for large-vocabulary
continuous speech recognition”, *ICASSP*;
Piotr Mirowski et al. (2010) “Feature-rich continuous
language models for speech recognition”, *SLT*]

- ❖ HUB-4 TV broadcast transcripts
- ❖ Re-rank 100-best lists returned by acoustic model
 - ❖ choose sentence with highest combined language model and acoustic model log-likelihoods (scores)

the american popular culture
americans popular culture
american popular culture
the nerds in popular culture
mayor kind popular culture
near can popular culture
the mere kind popular culture
...

Algorithm	Accuracy (test set)
Worst of 100-best list	57.8%
AT&T Watson system (acoustic and language models)	63.7%
smoothed 4-gram	63.5%
neural language model	64.1%
neural language model with part- of-speech tags and topic model	64.6%
Best of 100-best list (“Oracle”)	66.6%

Recurrent Neural Networks (RNN)

- ❖ Recurrent Neural Network (RNN) models

[Jeffrey L Elman (1991) “Distributed representations, simple recurrent networks and grammatical structure”, *Machine Learning*]

- ❖ **State variable** for arbitrarily long contexts

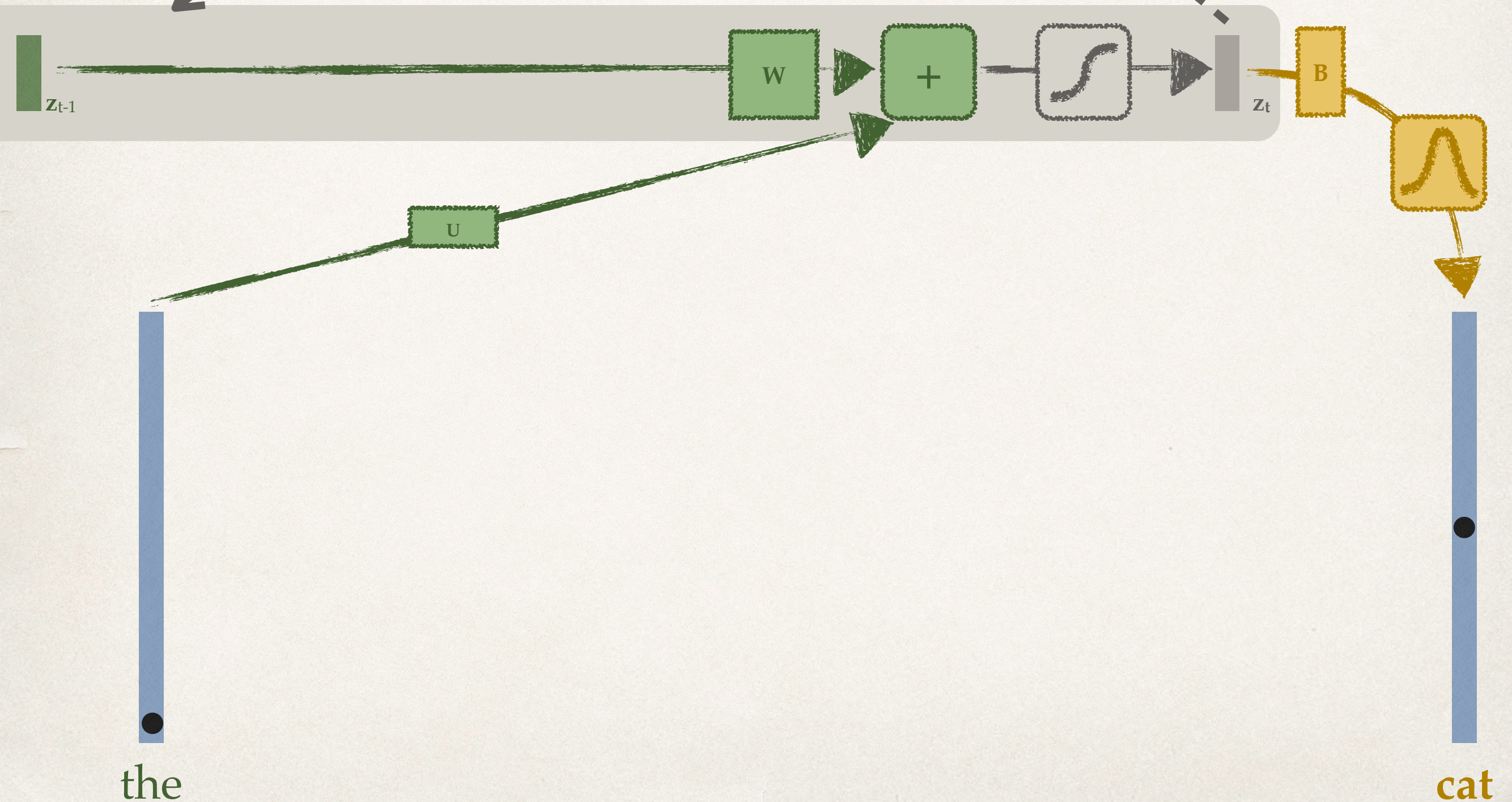
- ❖ “**persistent**” memory

- ❖ Recurrent neural language models

[Tomas Mikolov et al. (2010) “Recurrent neural network based language model”, *INTERSPEECH*]

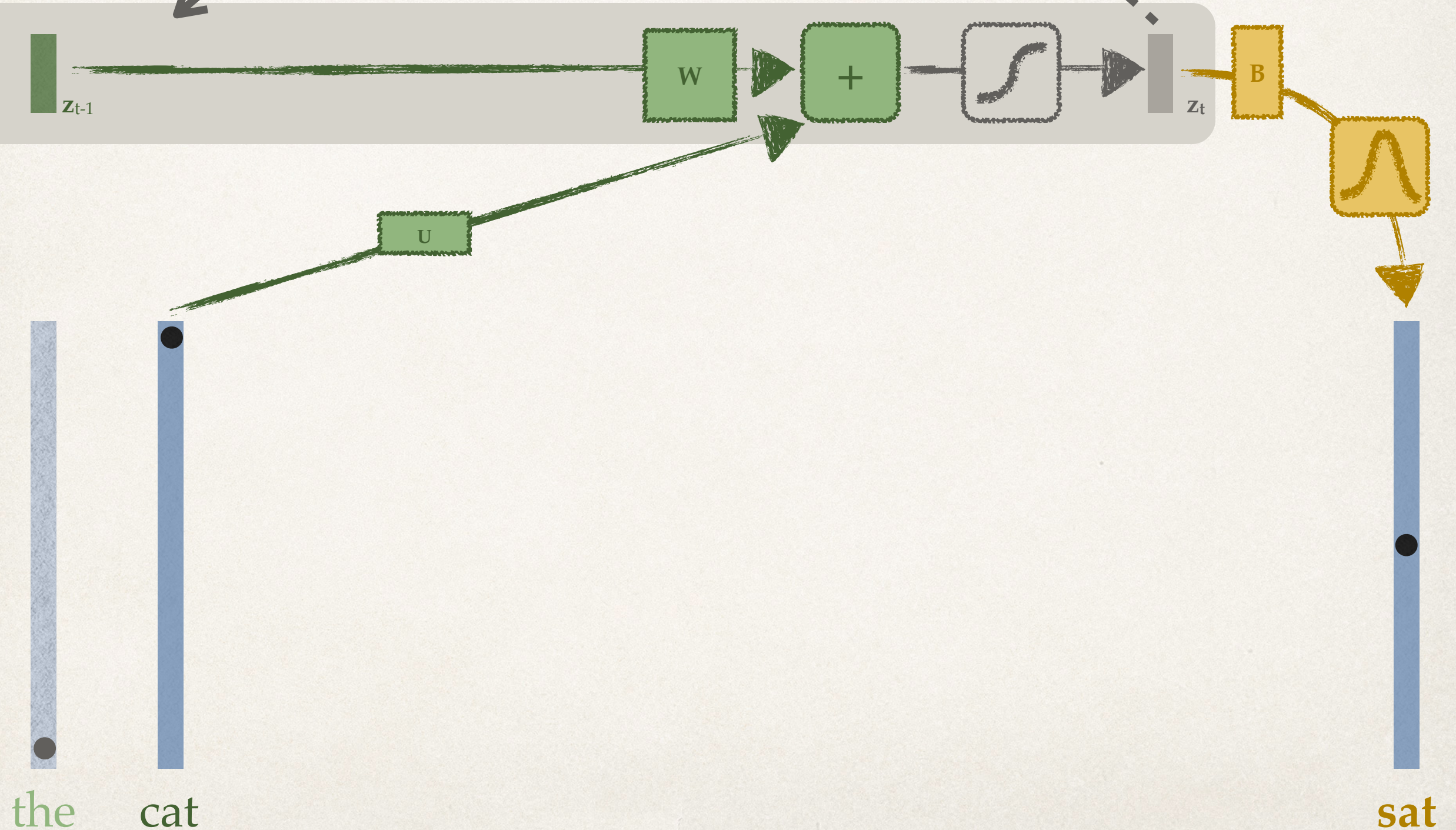
Recurrent Neural Network (RNN)

language models



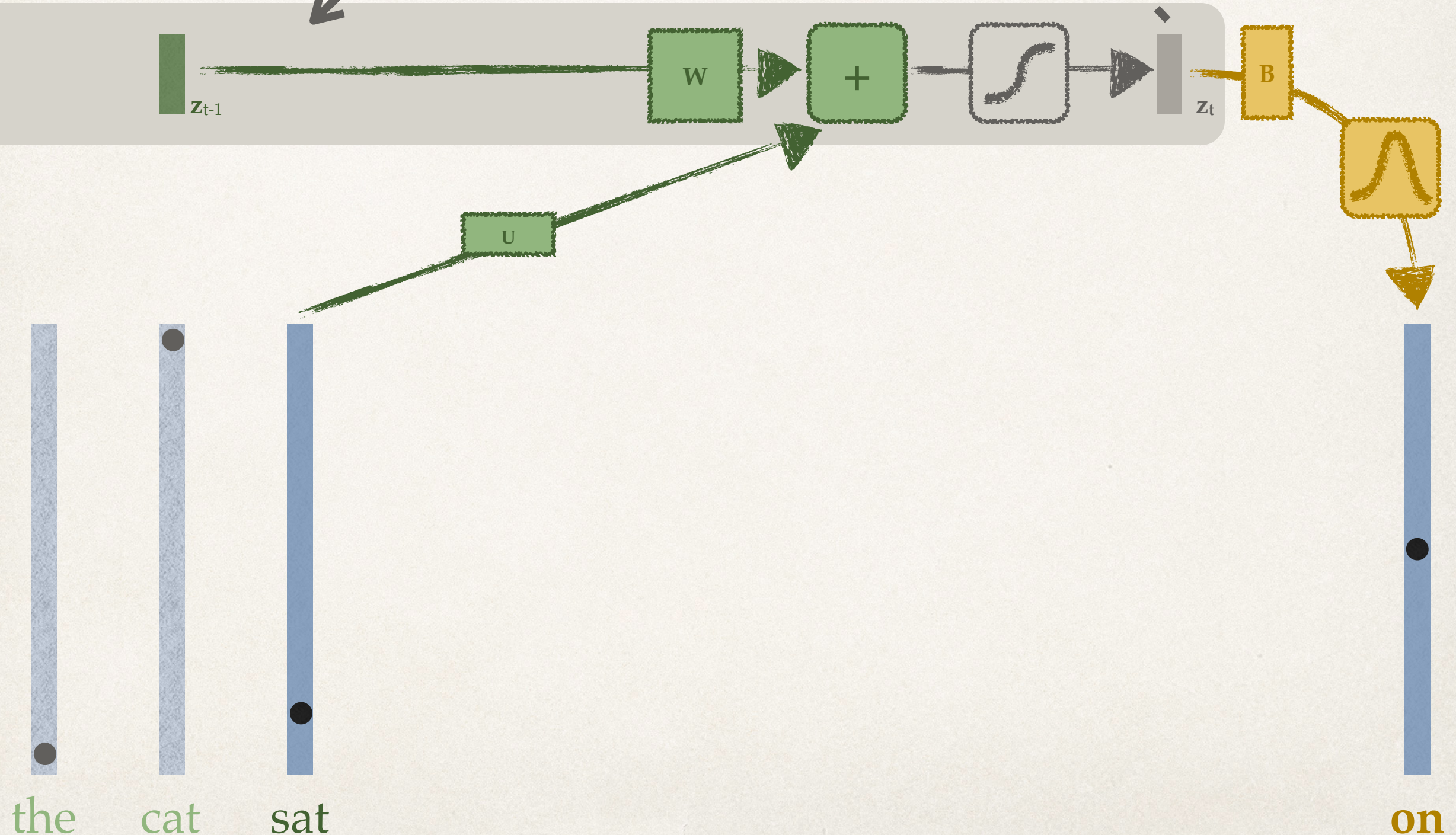
Recurrent Neural Network (RNN)

language models



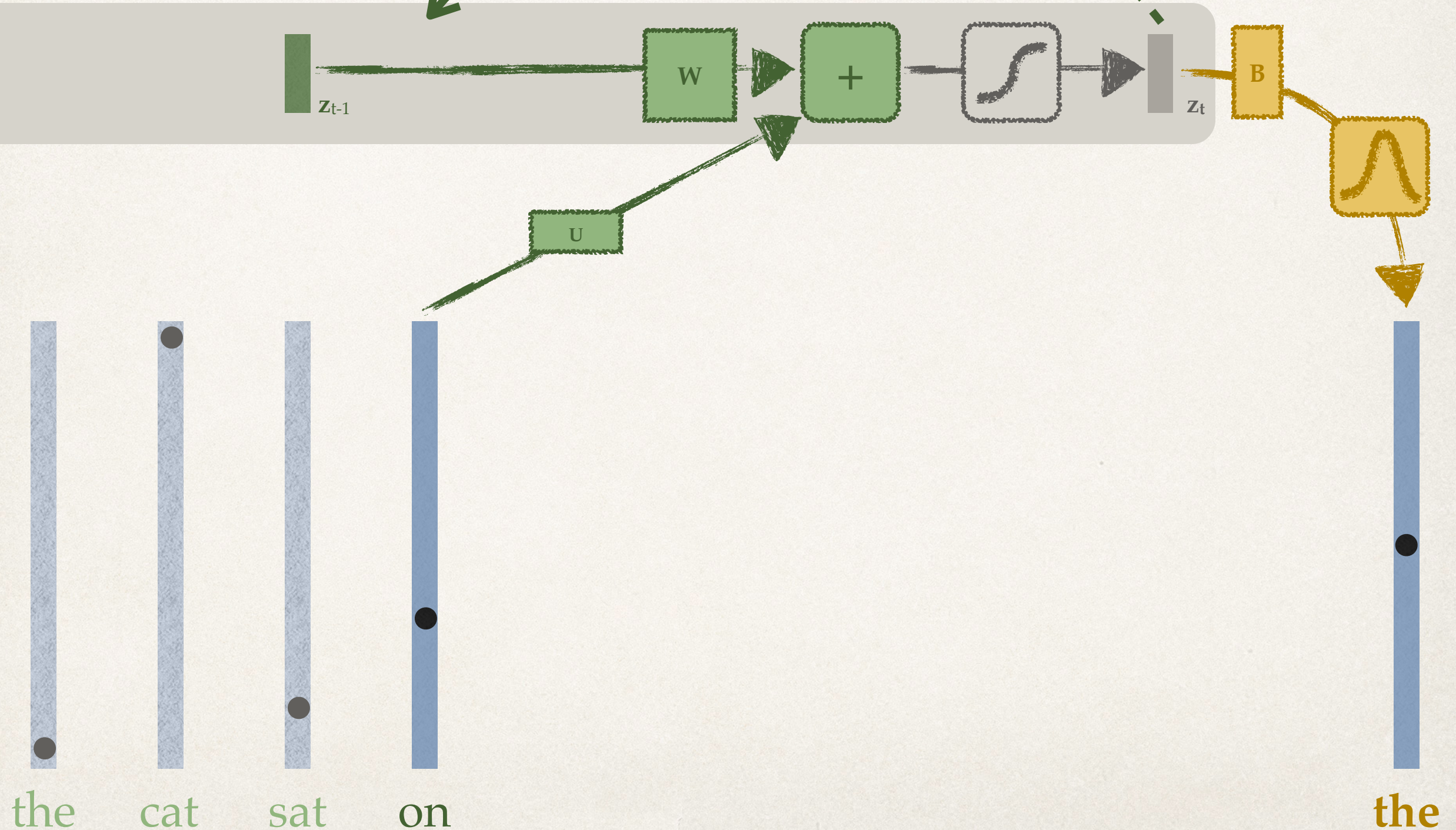
Recurrent Neural Network (RNN)

language models



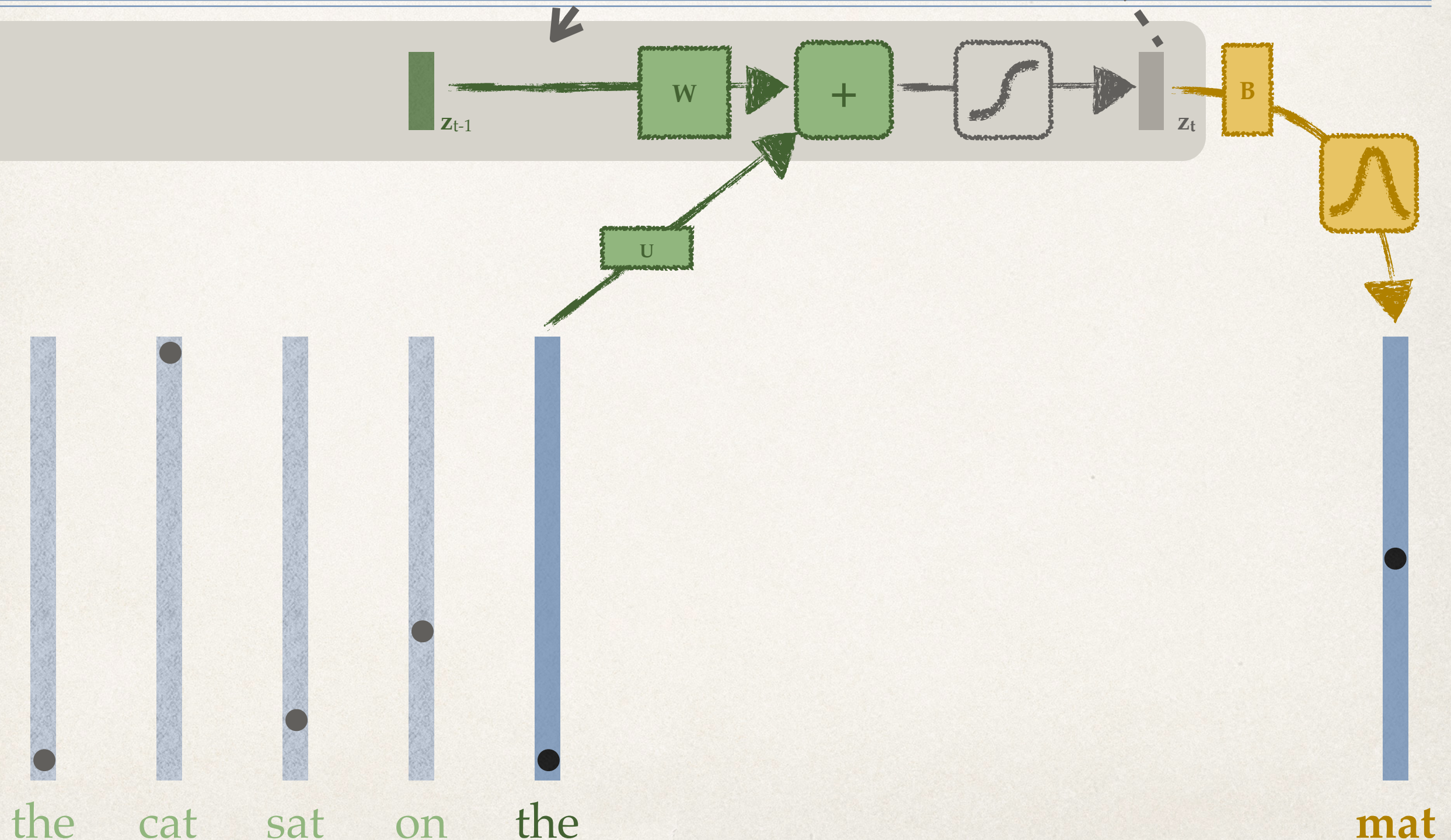
Recurrent Neural Network (RNN)

language models



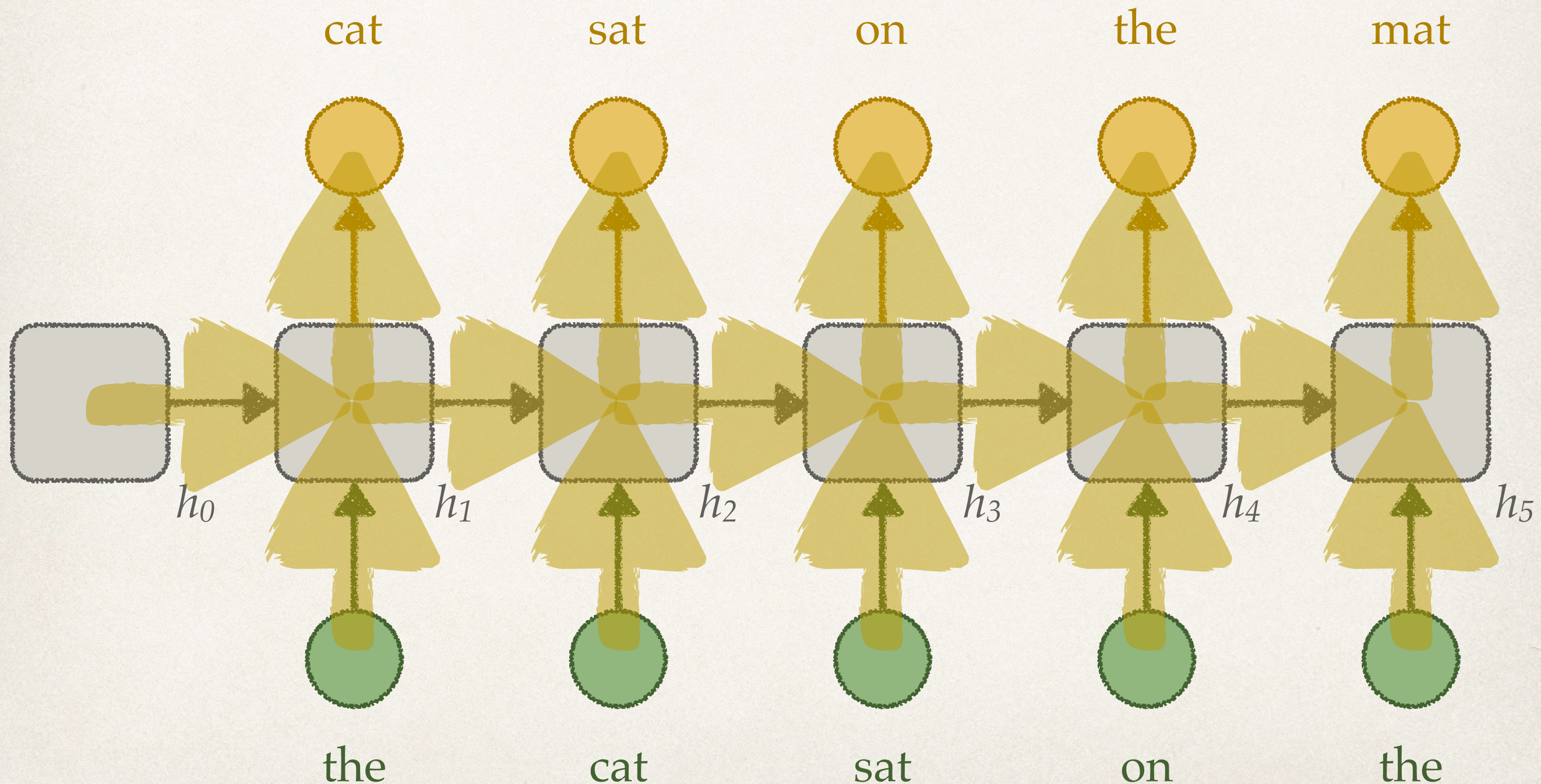
Recurrent Neural Network (RNN)

language models

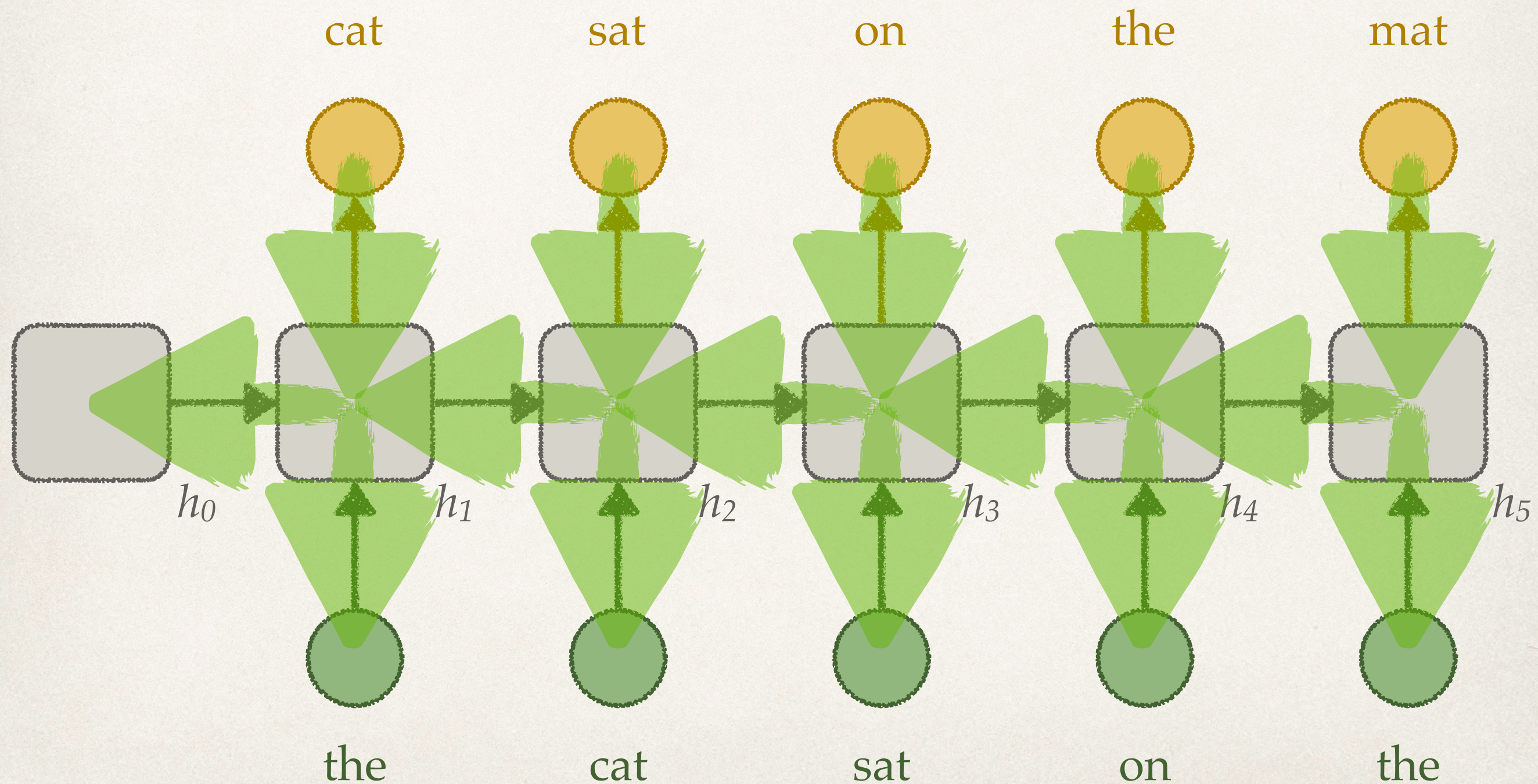


Recurrent Neural Network (RNN)

language models

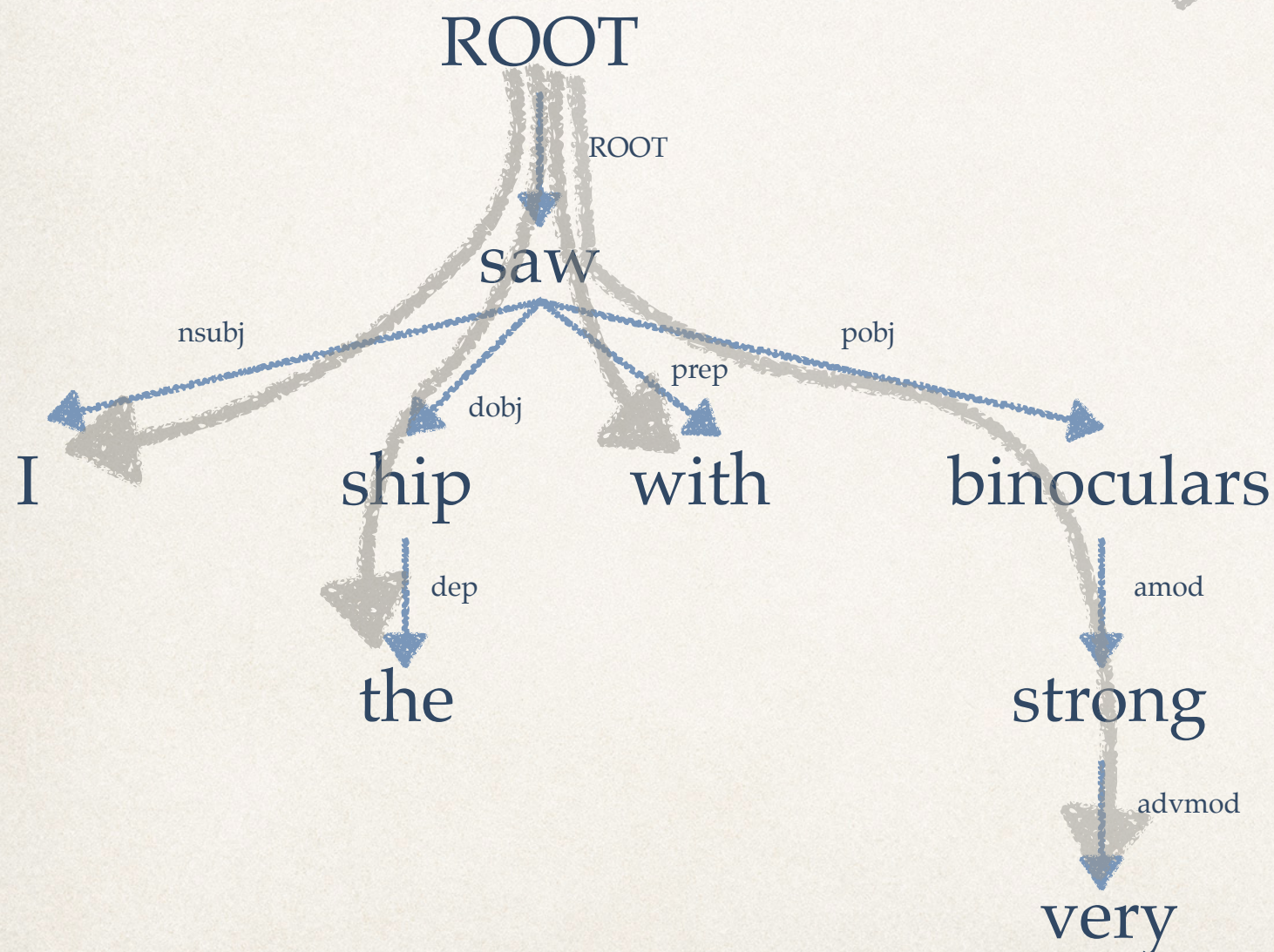


Back-propagation through time



Beyond sequential models: dependency parse tree-based RNN

I saw a ship with very strong binoculars



Algorithm	Accuracy (test set)
random	20%
SVD (word-paragraph)	49%
skip-gram	48%
smoothed 4-gram	39%
RNN + 4-gram features	45%
RNN on dependency tree	53%
Long Short-Term Memory	63%
human	90%

[Piotr Mirowski and Andreas Vlachos (2015) "Dependency recurrent neural language models for sentence completion", *ACL*;
Kai Sheng et al. (2015) "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", *ACL*;
Xiaodan Zhu et al. (2015) "Long Short-Term Memory Over Recursive Structures", *ICML*]

Table of contents

- ❖ **Representing words**

- ❖ Distributional Semantics
- ❖ Skip-grams and word2vec
- ❖ Sentence completion

- ❖ **Neural language models**

- ❖ N-grams and language modeling
- ❖ Recurrent Neural Networks (RNNs) and RNNLM
- ❖ Speech recognition

- ❖ **Recent developments**

- ❖ Long Short-Term Memory RNNs
- ❖ Sentence-to-sentence machine translation
- ❖ Image captioning



Motivation: end-to-end learning for natural language processing

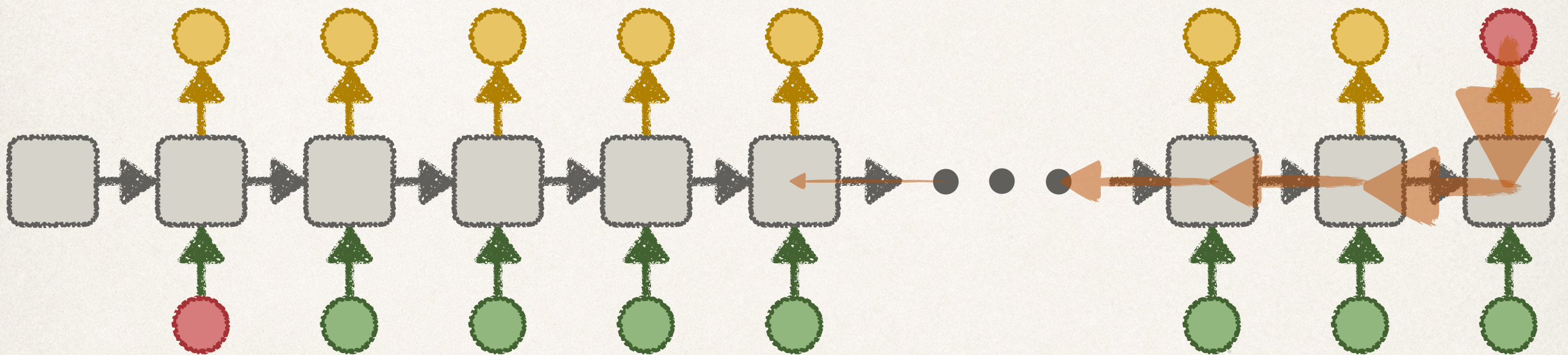
- ❖ One **integrated** algorithm for:
 - ❖ Speech recognition from **acoustic vectors** to **text**
 - ❖ Machine translation from **one language** to **another**
 - ❖ Image captioning from **image** to **text**



[Image credits: Vinyals et al (2014)]

Learning long-range dependencies in RNNs is difficult

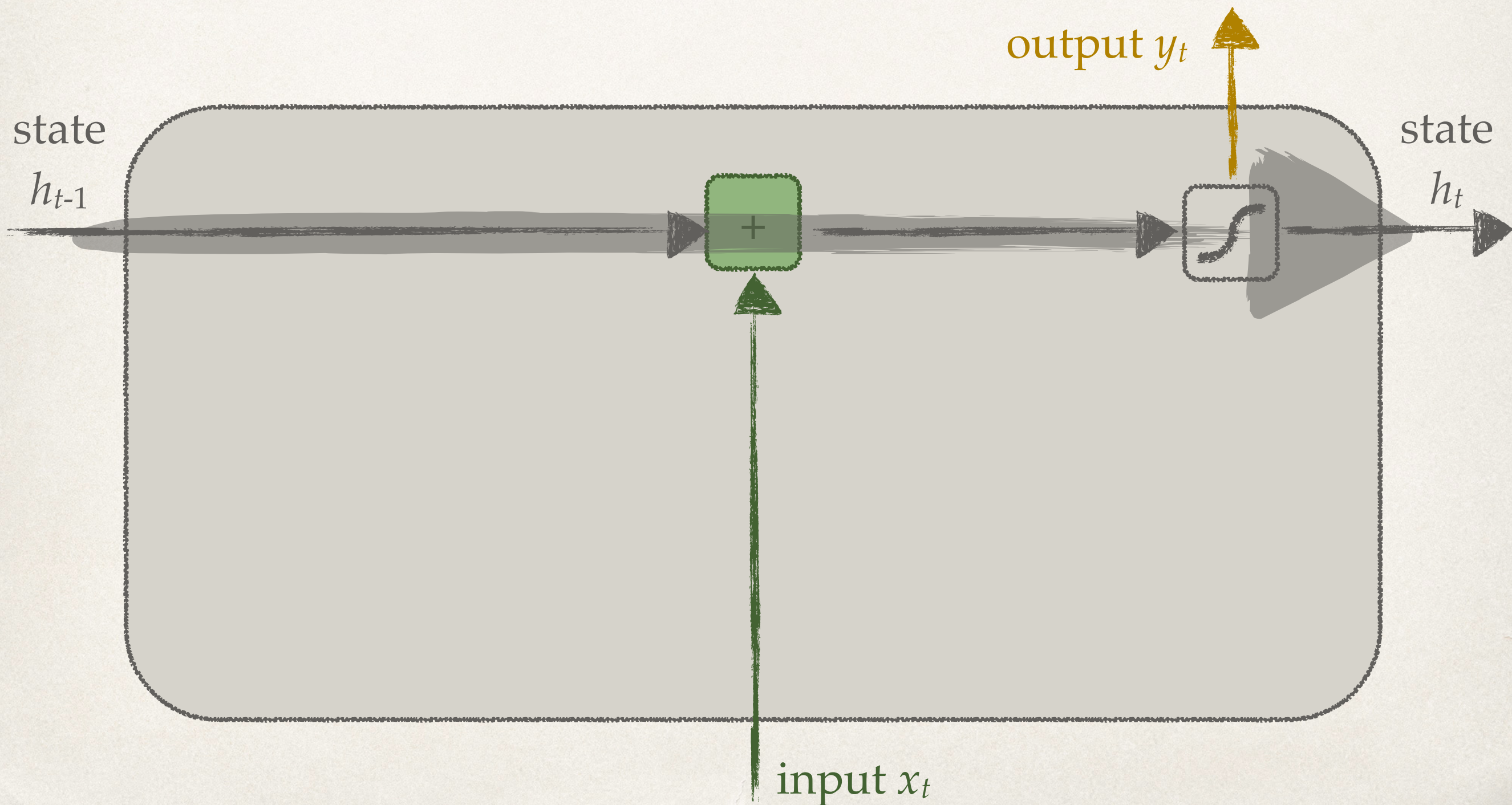
(n -grams cannot retain information beyond n steps)



Because of the **non-linearity** in the hidden units,
gradients of the error during back-propagation
decay **exponentially** with the length of the sequence

[Sepp Hochreiter (1991) "Untersuchungen zu dynamischen neuronalen Netzen", *Diploma TUM*;
Yoshua Bengio et al. (1994) "Learning Long-Term Dependencies with Gradient Descent is Difficult"
IEEE Transactions on Neural Networks]

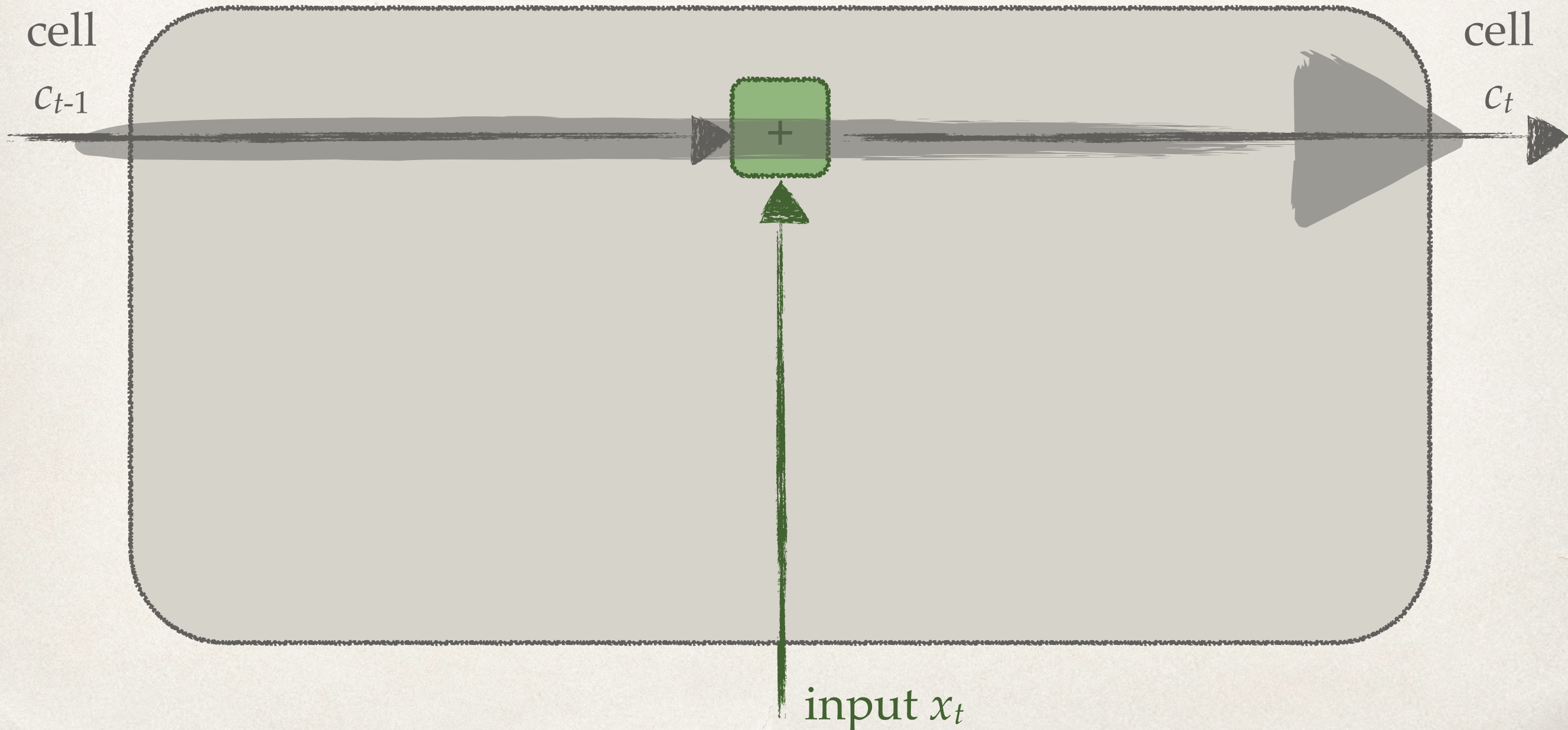
Recurrent Neural Networks



Long Short-Term Memory (LSTM)

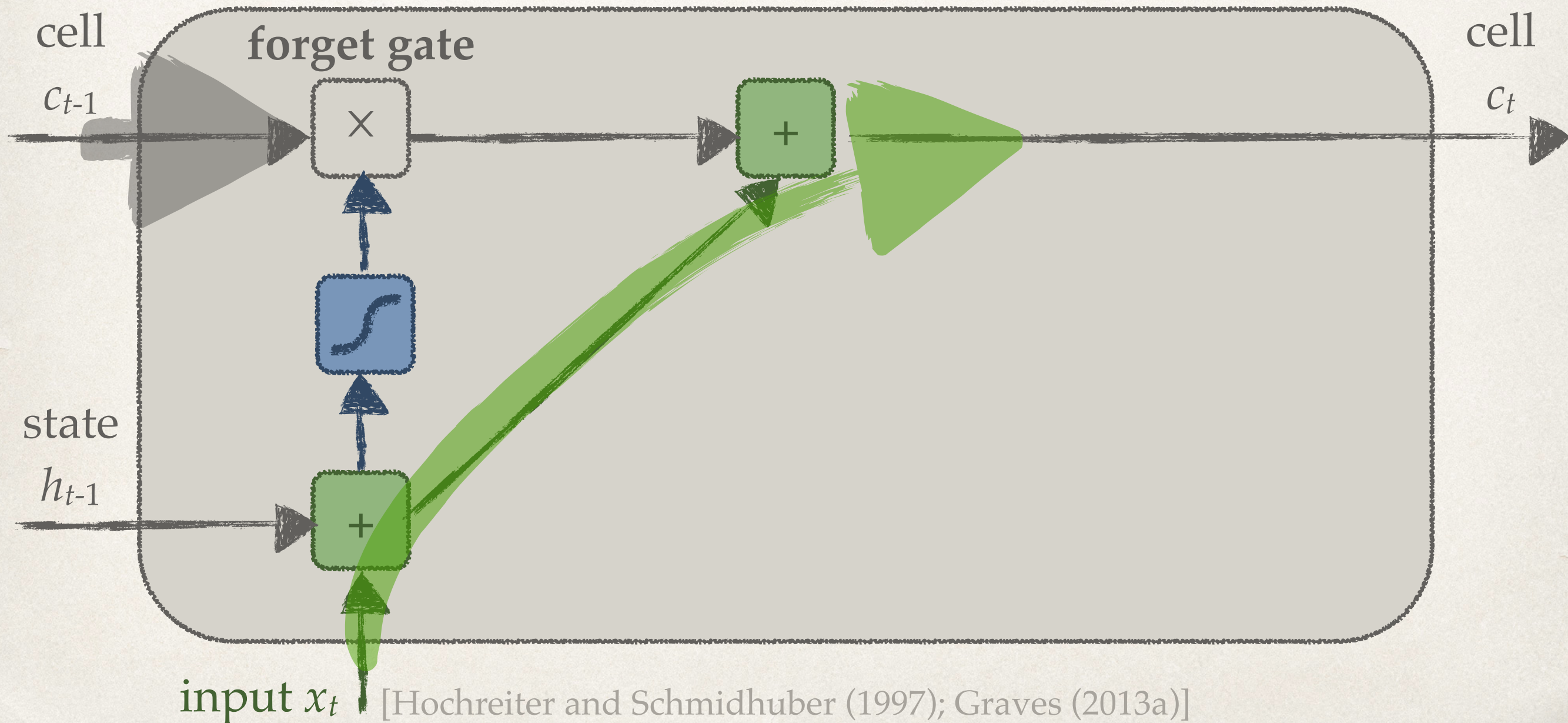
Requirement #1: linear cell

[Sepp Hochreiter and Jürgen Schmidhuber (1997) "Long Short-Term Memory", *Neural Computation*;
Alex Graves (2013a) "Generating sequences with recurrent neural networks", *arXiv* 1308.0850]



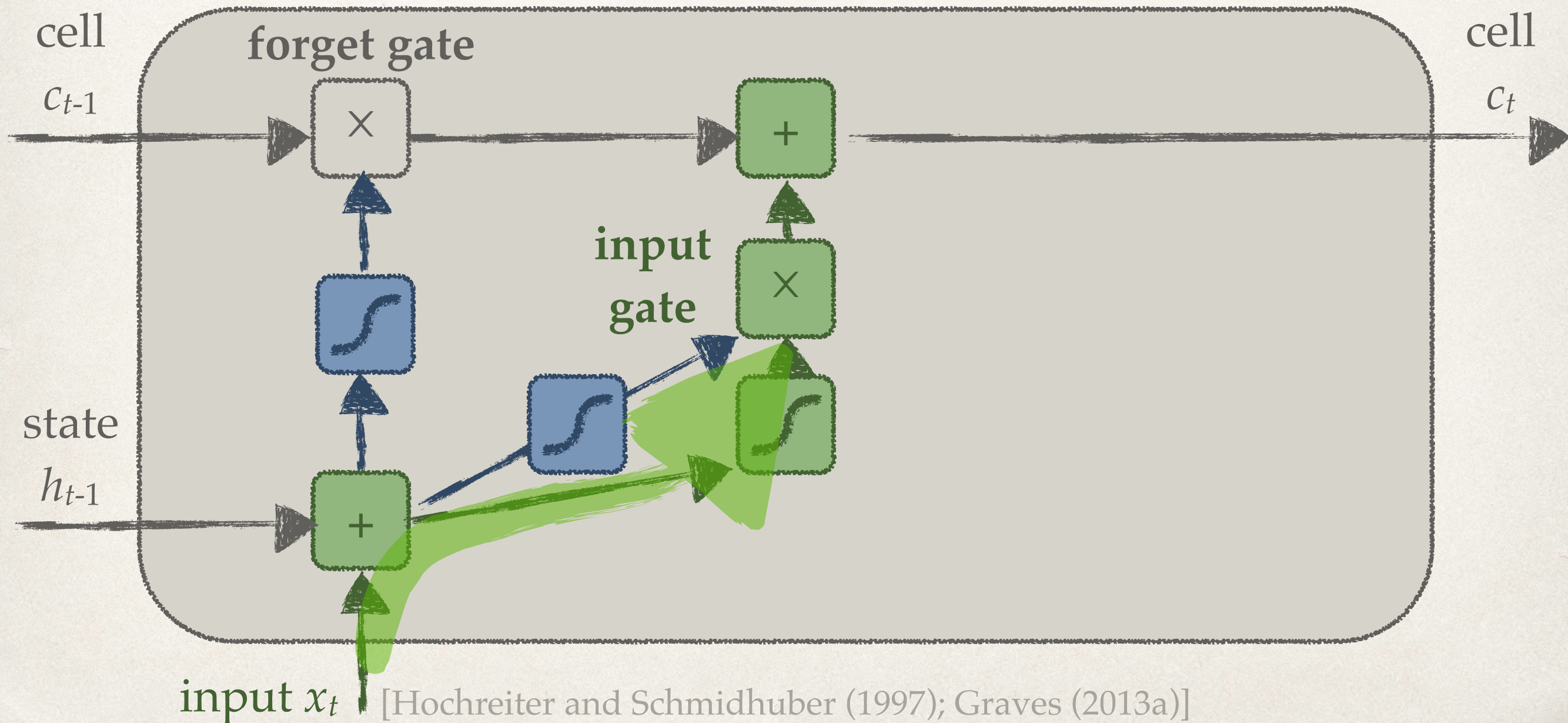
Long Short-Term Memory (LSTM)

Requirement #2: forget information



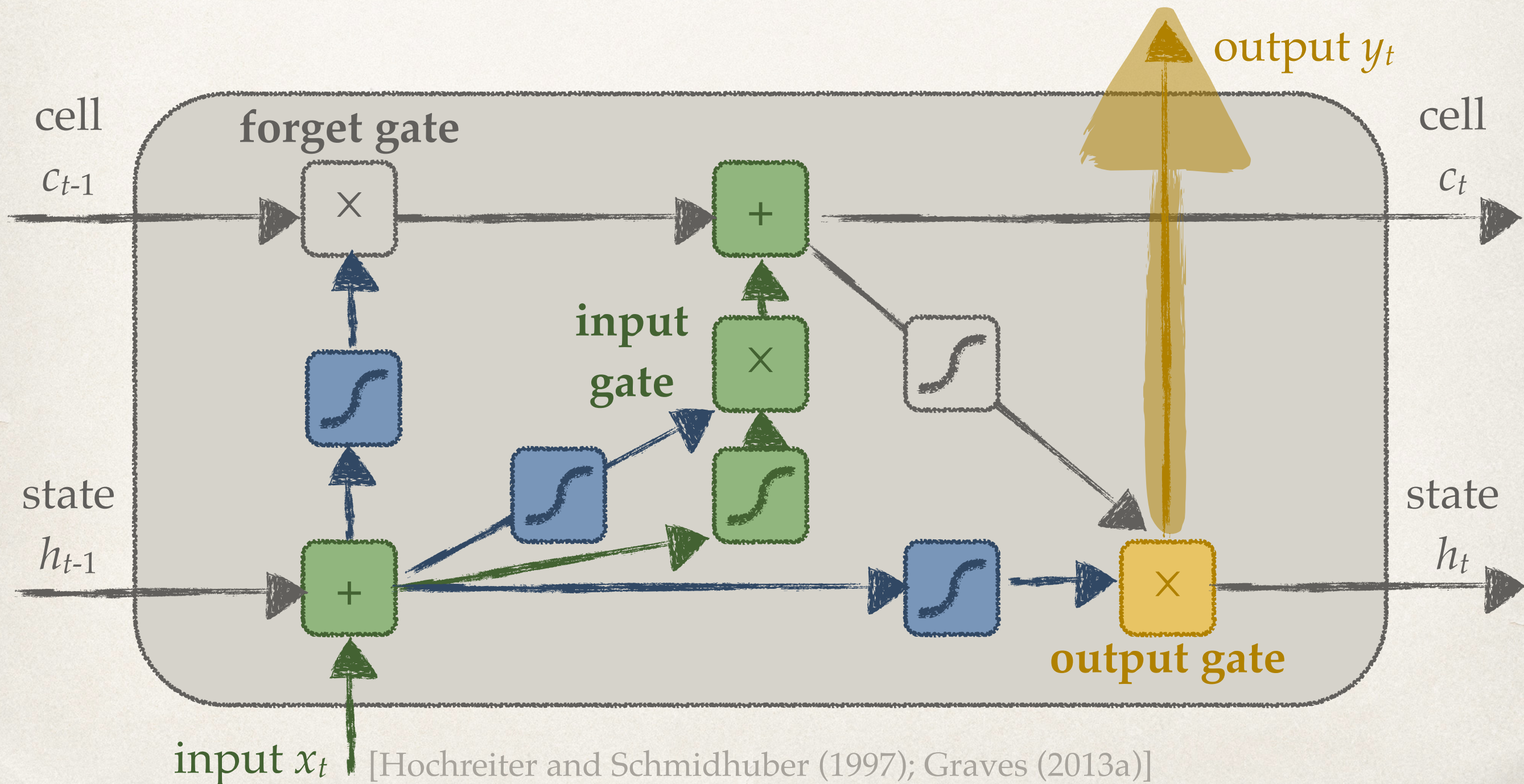
Long Short-Term Memory (LSTM)

Requirement #3: ignore inputs

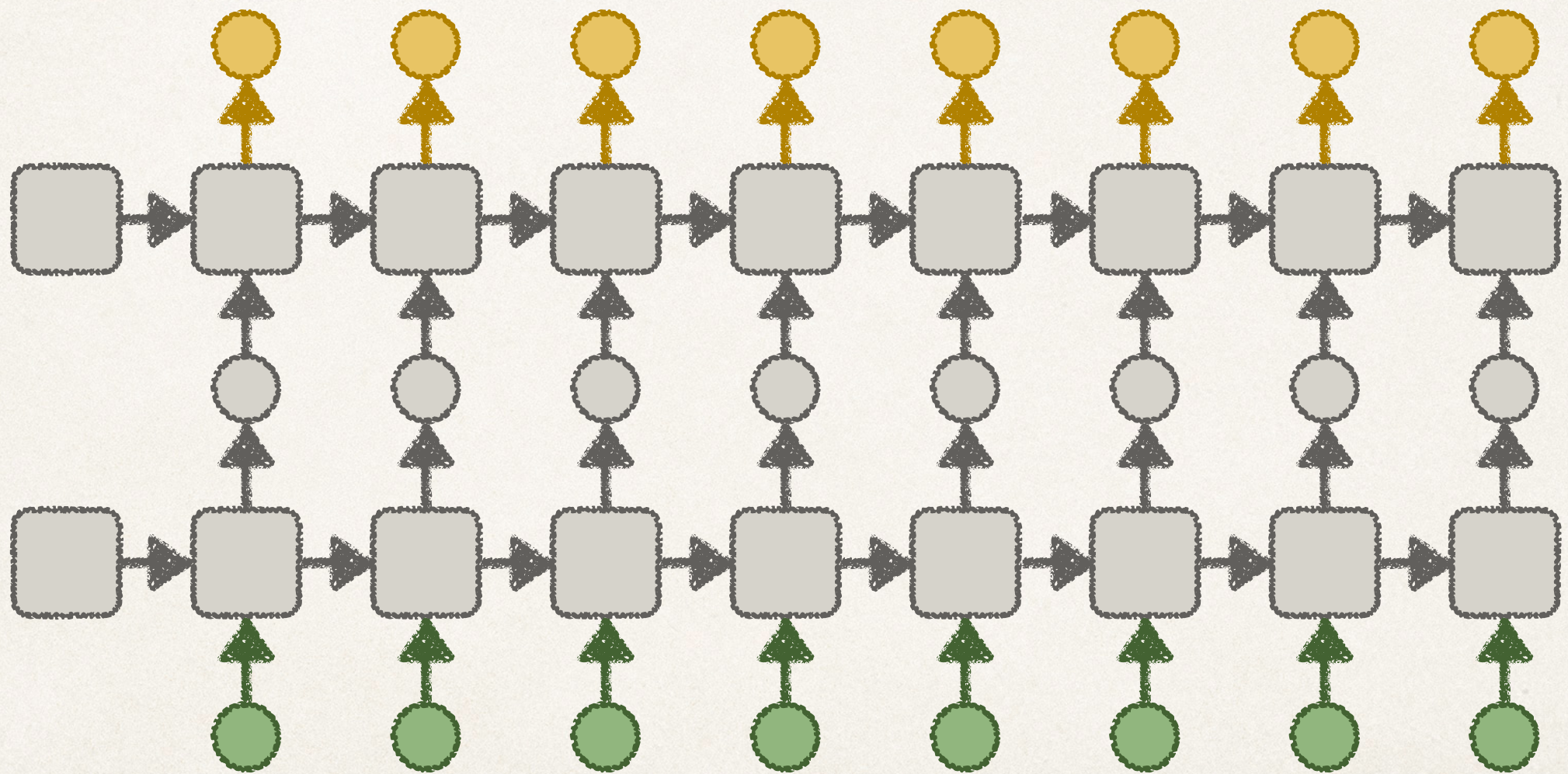


Long Short-Term Memory (LSTM)

Requirement #4: control output



Deep LSTMs: stacking layers



[Graves (2013a)]

Sentence-to-sentence machine translation

[Sutskever et al. (2014)]

“Sequence to sequence learning with neural networks”, *NIPS*]

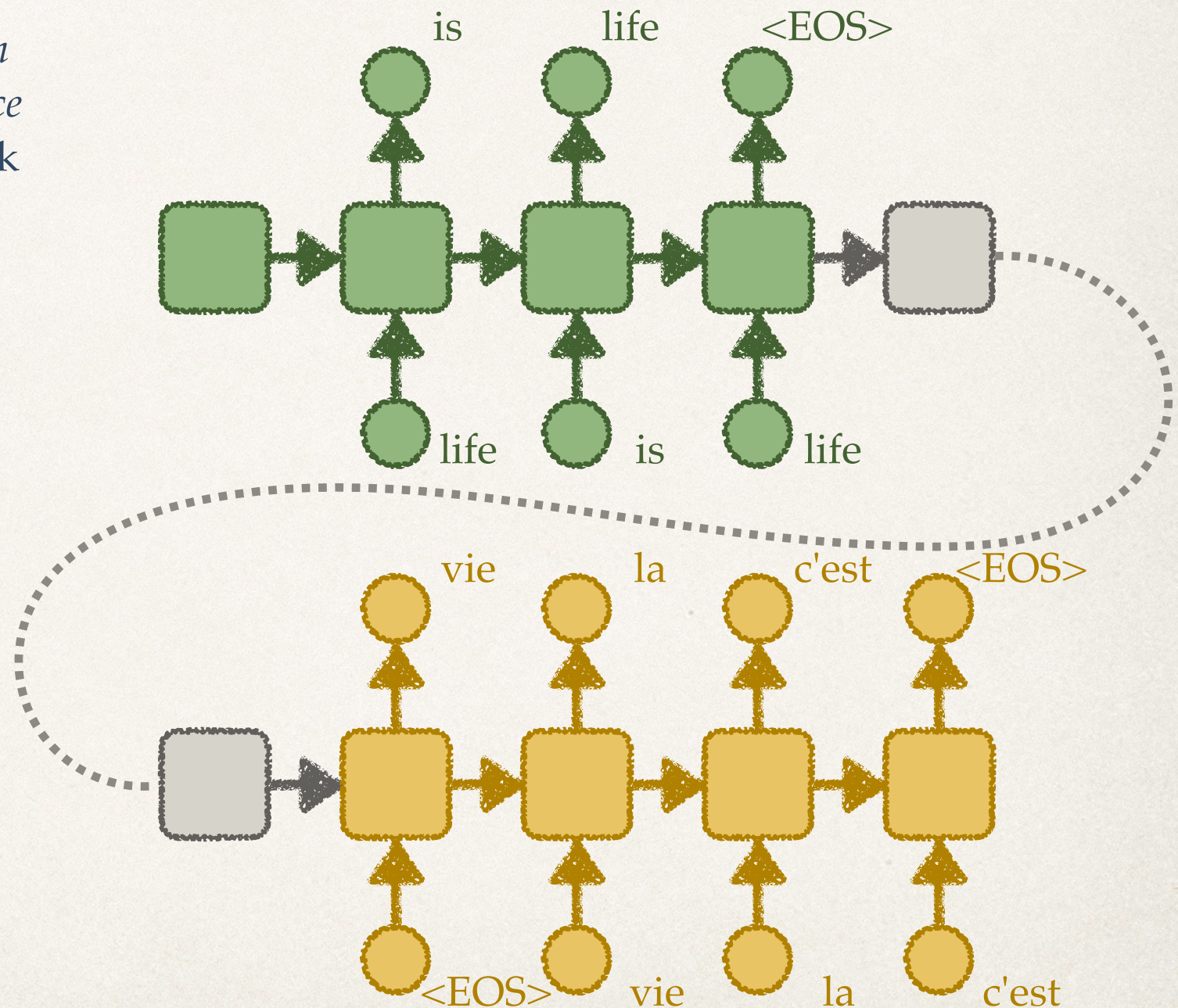
English
sentence
 $V_E=160k$

French
sentence
 $V_F=80k$

4 layers
 $D=1000$

LSTM
encoder

LSTM
decoder



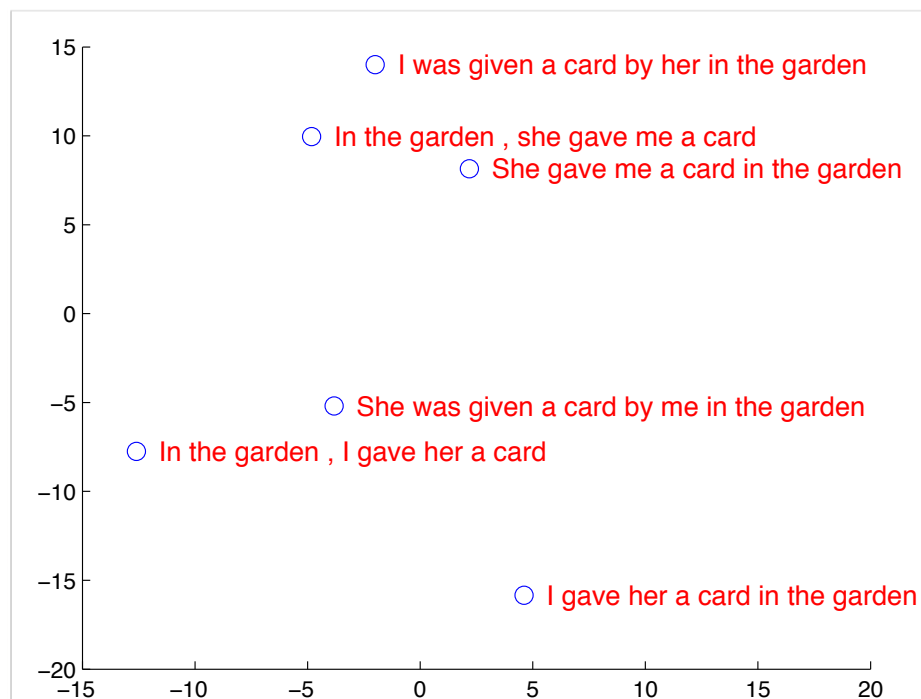
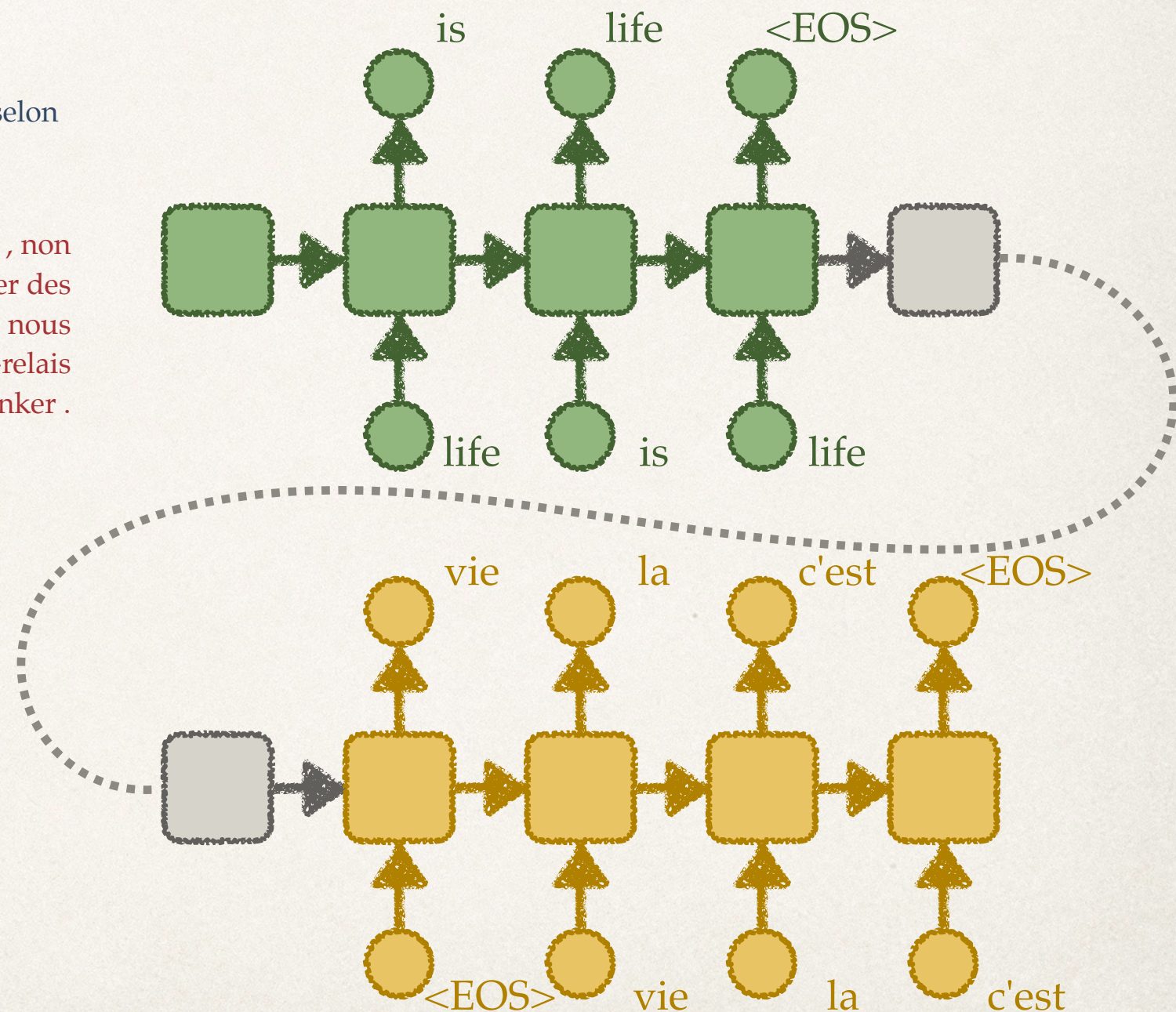
Sentence-to-sentence machine translation

[Sutskever et al. (2014)]

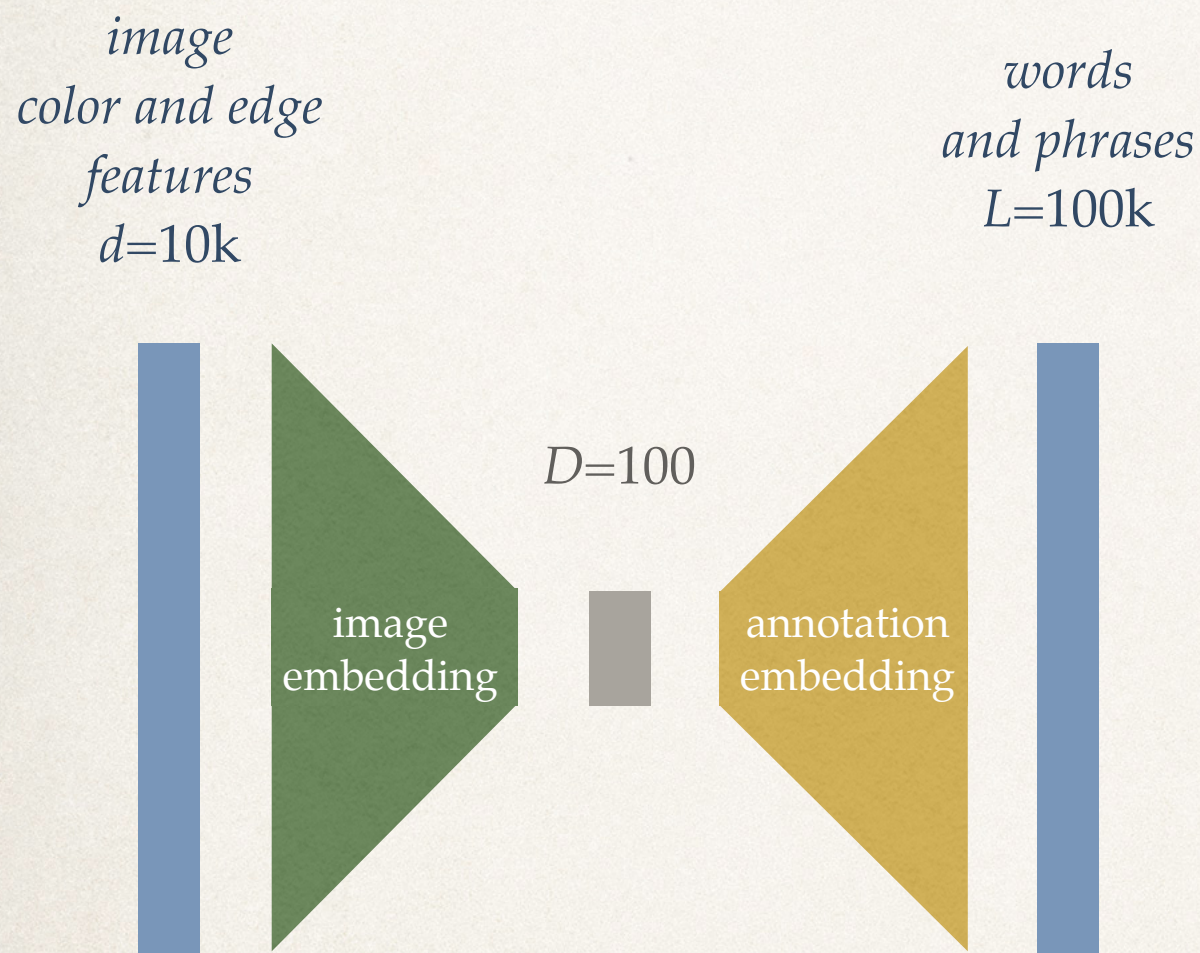
“Sequence to sequence learning with neural networks”, *NIPS*]

“ Les téléphones cellulaires , qui sont vraiment une question , non seulement parce qu’ ils pourraient potentiellement causer des interférences avec les appareils de navigation , mais nous savons , selon la FCC , qu’ ils pourraient interférer avec les tours de téléphone cellulaire lorsqu’ ils sont dans l’ air “ , dit <UNK> .

“ Les téléphones portables sont véritablement un problème , non seulement parce qu’ ils pourraient éventuellement créer des interférences avec les instruments de navigation , mais parce que nous savons , d’ après la FCC , qu’ ils pourraient perturber les antennes-relais de téléphonie mobile s’ ils sont utilisés à bord “ , a déclaré Rosenker .





Large-scale web image annotation



(2.5M ImageNet + 10M Web) images

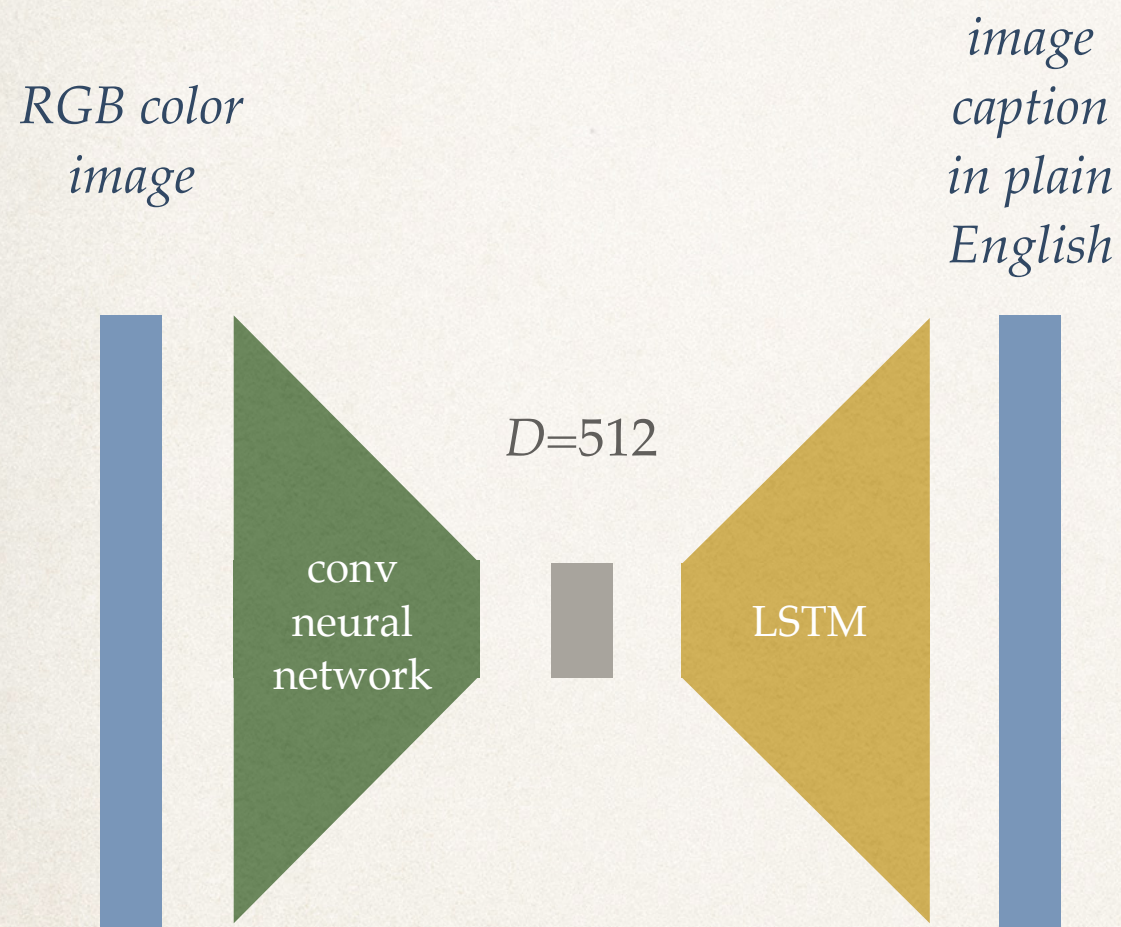
[Jason Weston et al (2010) "Large scale image annotation:
learning to rank with joint word-image embeddings",
Machine Learning]

Image	WSABIE
	delfini, <i>orca</i> , dolphin , mar, delfin, dauphin, <i>whale</i> , can-cun, <i>killer whale</i> , sea world
	eiffel tower , <i>statue</i> , <i>eiffel</i> , mole an-toneliana, la tour eiffel, londra, cctv tower, <i>big ben</i> , cala-trava, <i>tokyo tower</i>

[Image credits: Weston et al (2010)]

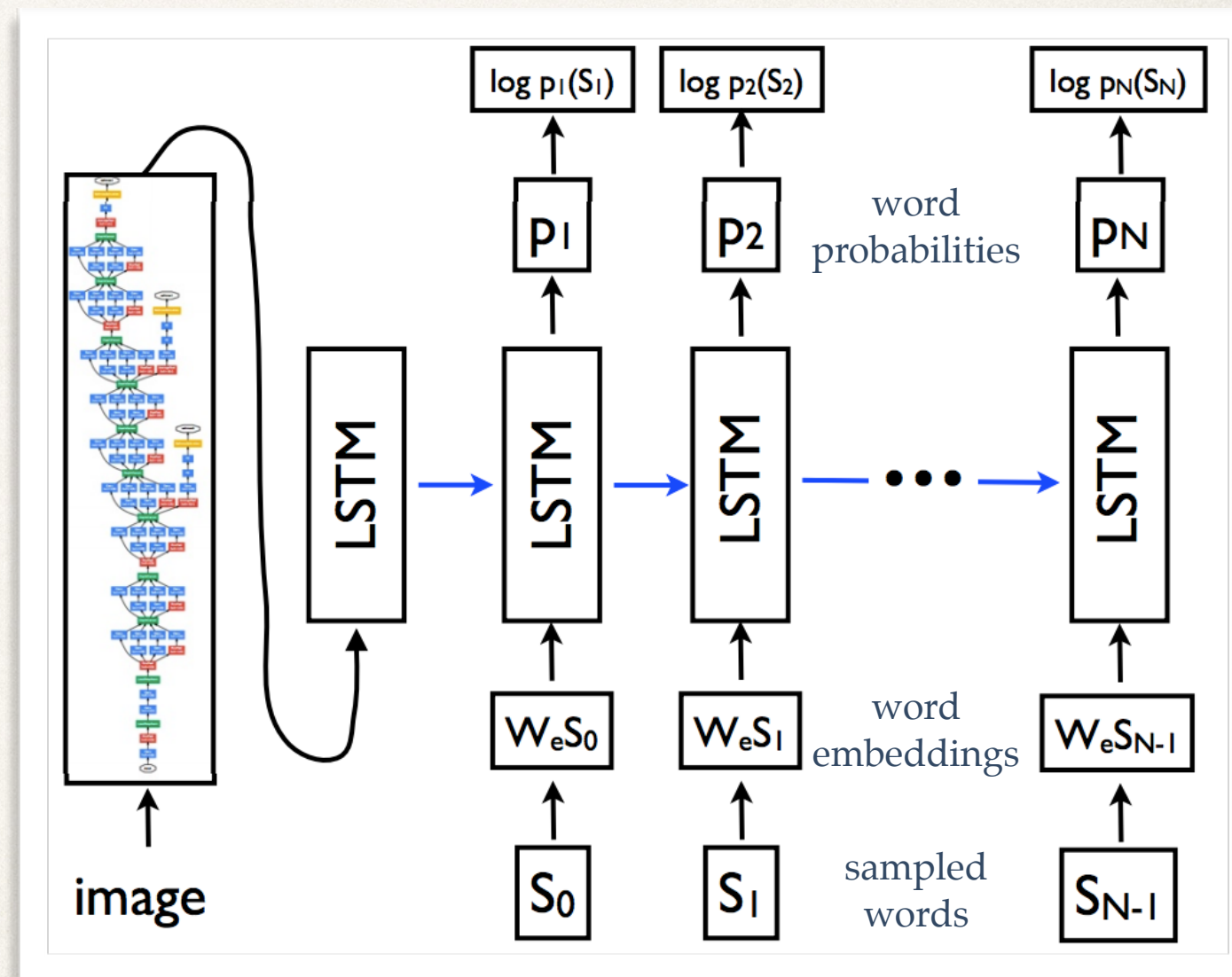
Image captioning

[Vinyals et al. (2014) "Show and Tell: Neural Image Caption Generation";
Karpathy et al. (2014) "Deep Visual-Semantic Alignments for Generating Image Descriptions";
Kiros et al. (2014) "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models"]



convolutional network
pre-trained on 2.5M ImageNet images

end-to-end system trained on 100k to 1M
image - caption pairs



[Image credits: Vinyals et al. (2014) "Show and Tell: Neural Image Caption Generation"]

Image captioning

[Vinyals et al. (2014) "Show and Tell: Neural Image Caption Generation";

Karpathy et al. (2014) "Deep Visual-Semantic Alignments for Generating Image Descriptions";

Kiros et al. (2014) "Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models"]

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

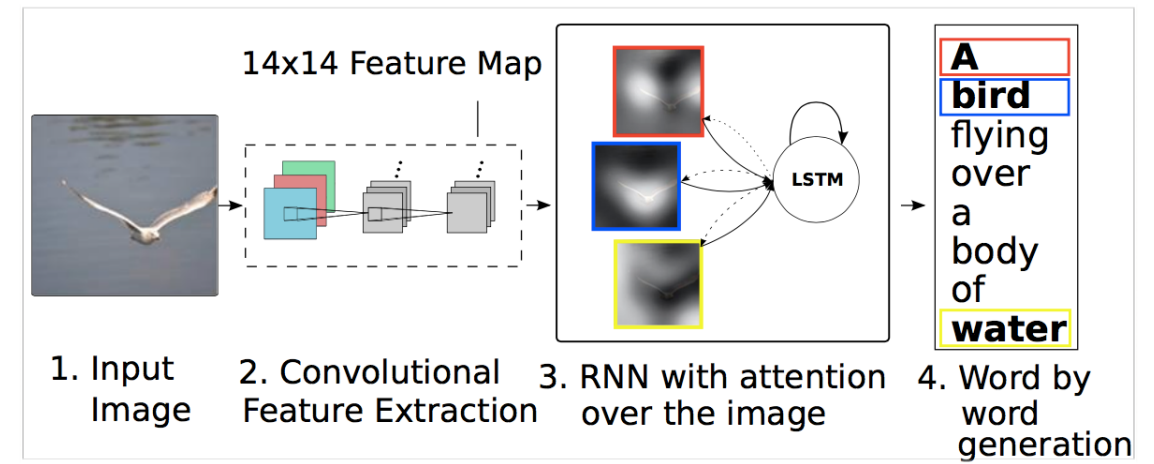
Describes with minor errors

Somewhat related to the image

Unrelated to the image

[Image credits: Vinyals et al. (2014) "Show and Tell: Neural Image Caption Generation"]

Image captioning with visual attention



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.

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

Answering queries given context, using attention mechanism

by *ent423* ,*ent261* correspondent updated 9:49 pm et ,thu
march 19 ,2015 (*ent261*) a *ent114* was killed in a parachute
accident in *ent45* ,*ent85* ,near *ent312* ,a *ent119* official told
ent261 on wednesday .he was identified thursday as
special warfare operator 3rd class *ent23* ,29 ,of *ent187* ,
ent265 .` *ent23* distinguished himself consistently
throughout his career .he was the epitome of the quiet
professional in all facets of his life ,and he leaves an
inspiring legacy of natural tenacity and focused

. . .

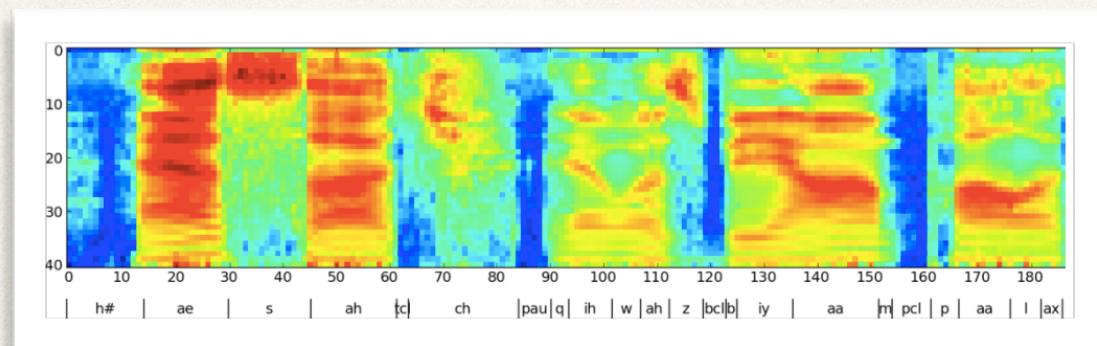
ent119 identifies deceased sailor as **X** ,who leaves behind
a wife

[Karl M Hermann et al. (2015) "Teaching
Machines to Read and to Comprehend", *NIPS*]

A few exciting applications of LSTM (not covered in this talk)

Speech recognition from acoustic vectors

[Graves et al. (2013b) "Speech recognition with deep recurrent neural networks", *ICASSP*]



Handwritten text generation

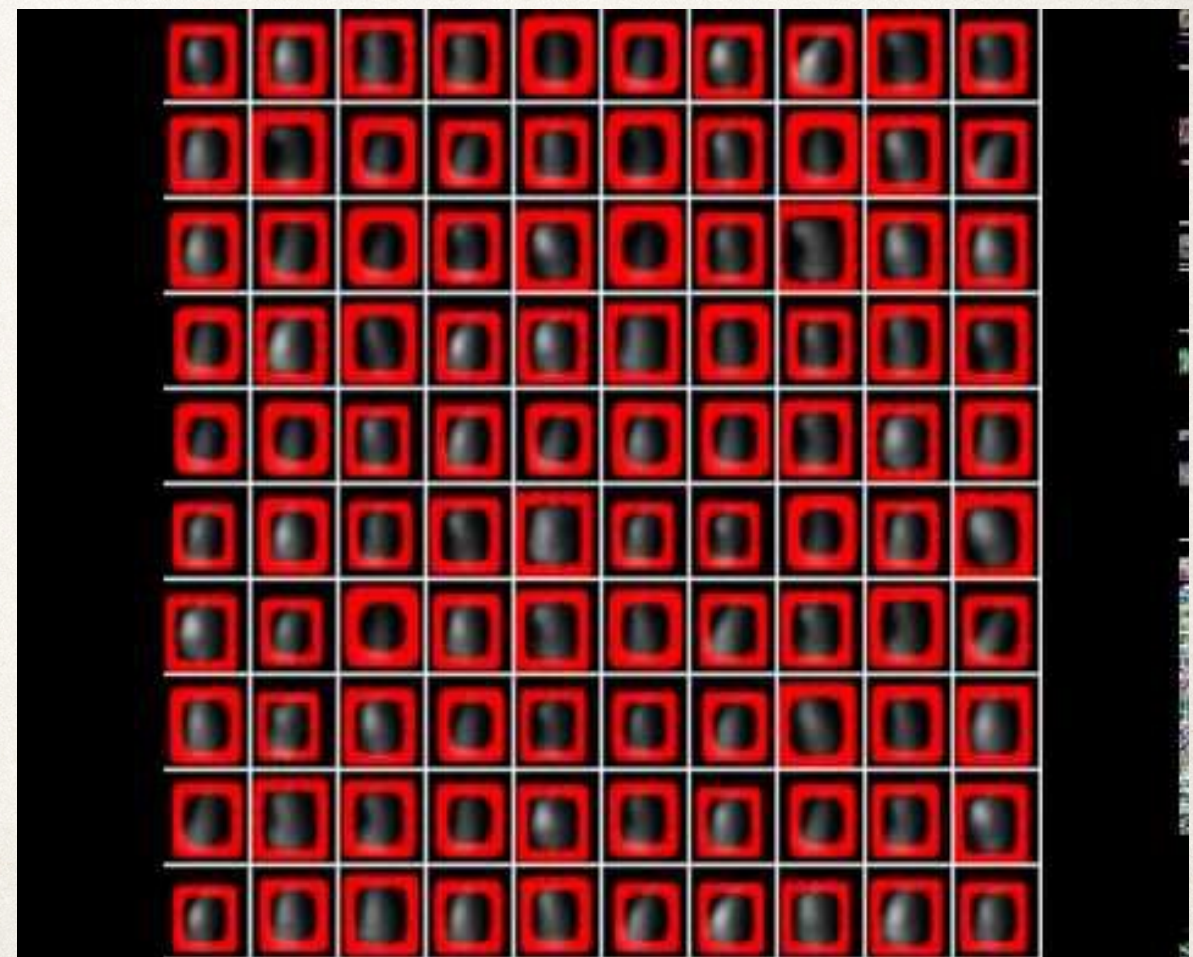
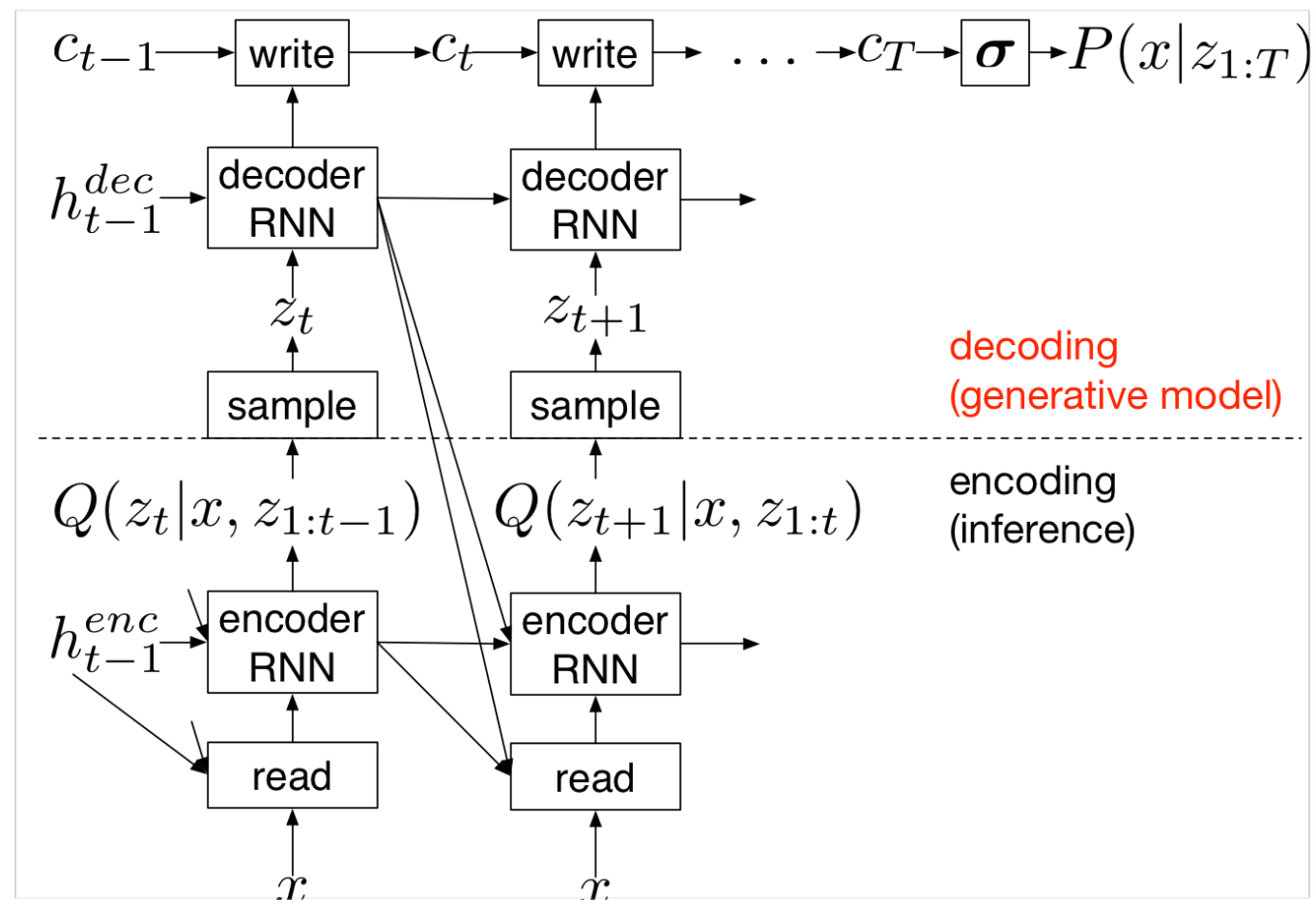
[Alex Graves (2013a) "Generating sequences with recurrent neural networks", *arXiv*]

from his travels it might have been
from his travels it might have been
from his travels it might have been
from his travels it might have been

DRAW: Deep Recurrent Attentive Writer

[Karol Gregor et al. (2015)
“DRAW: A recurrent
neural network for image
generation”, *ICML*]

Attention mechanism:
where to look
where to write



DRAW: Deep Recurrent Attentive Writer

[Karol Gregor et al. (2015)
“DRAW: A recurrent
neural network for image
generation”, *ICML*]



Thank you!

- ❖ piotrmiroski@google.com
- ❖ www.deepmind.com
- ❖ Matlab code for neural language models:
<https://github.com/piotrmiroski/LBL>
- ❖ C++ code for RNNs with dependency tree parsing:
<https://github.com/piotrmiroski/DependencyTreeRnn>
- ❖ These slides on:
<https://piotrmiroski.wordpress.com/>