What do our models learn?

Trying to Understand Neural Models for Language Processing

Yoav Goldberg

NLPL Winter School 2020

Bar-Ilan University

BIU NLP

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

How do we do NLP?

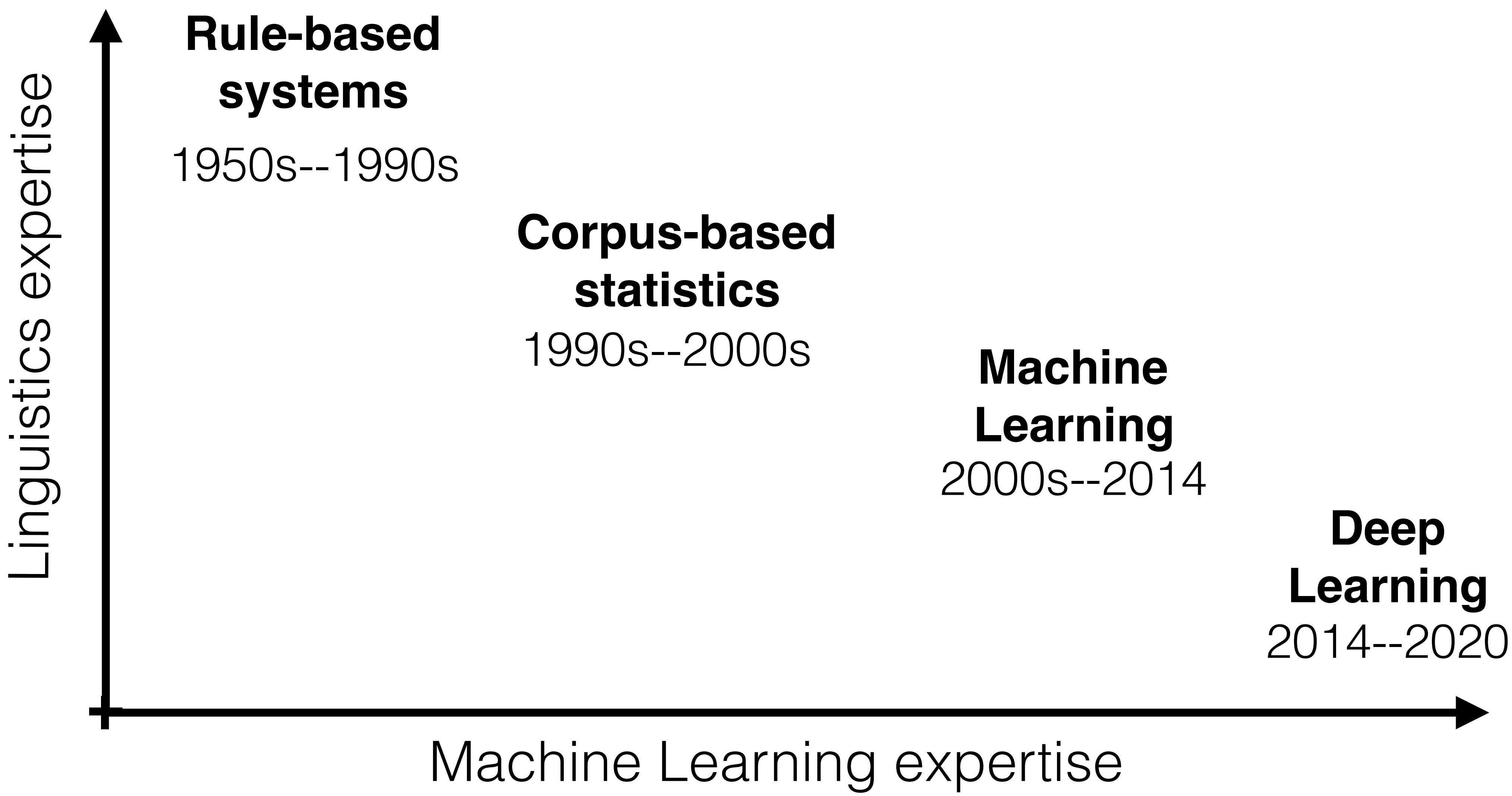
Rule-based systems
1950s--1990s

Corpus-based statistics
1990s--2000s

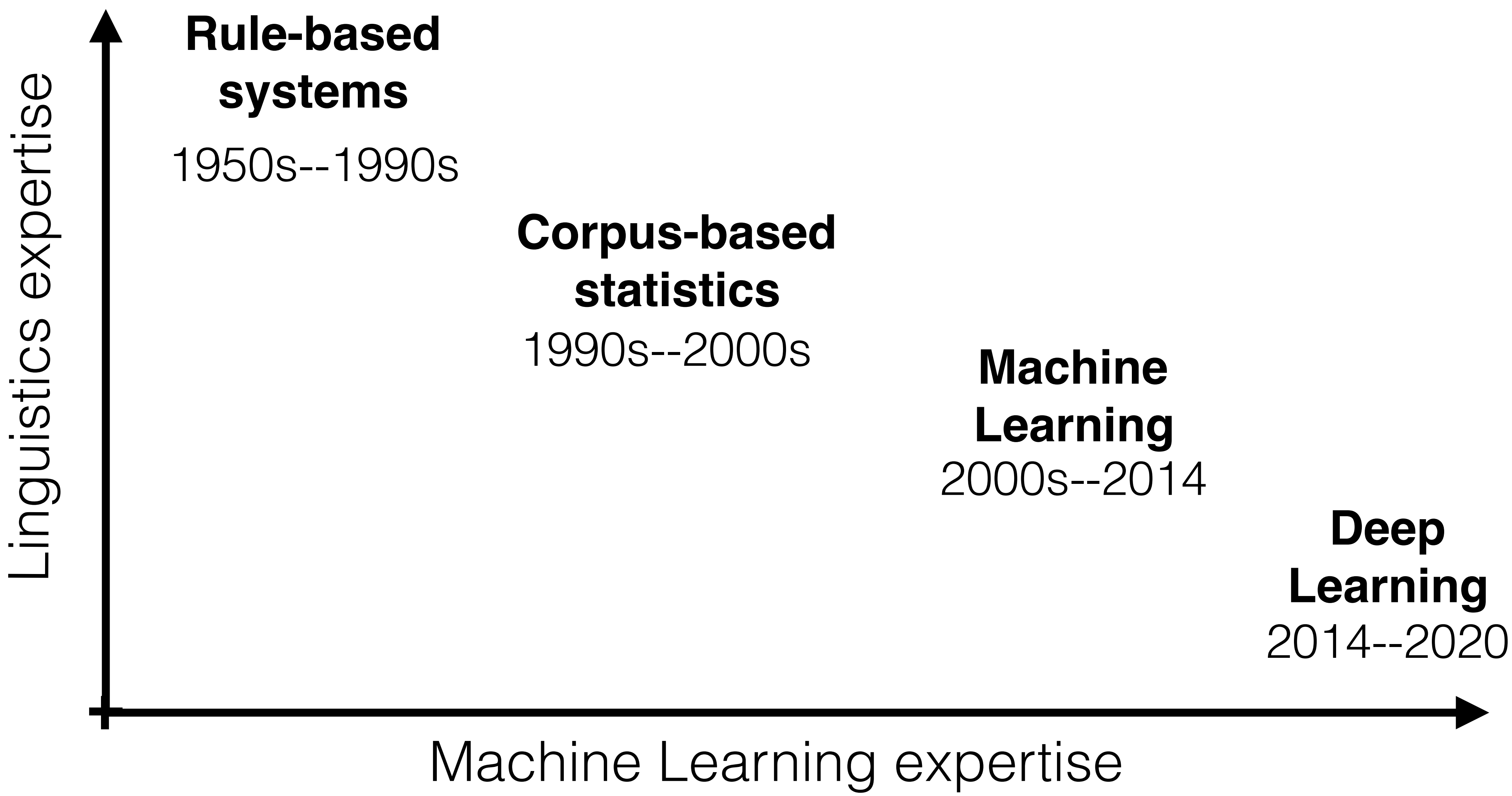
Machine Learning
2000s--2014

Deep Learning
2014--2020

How do we do NLP?

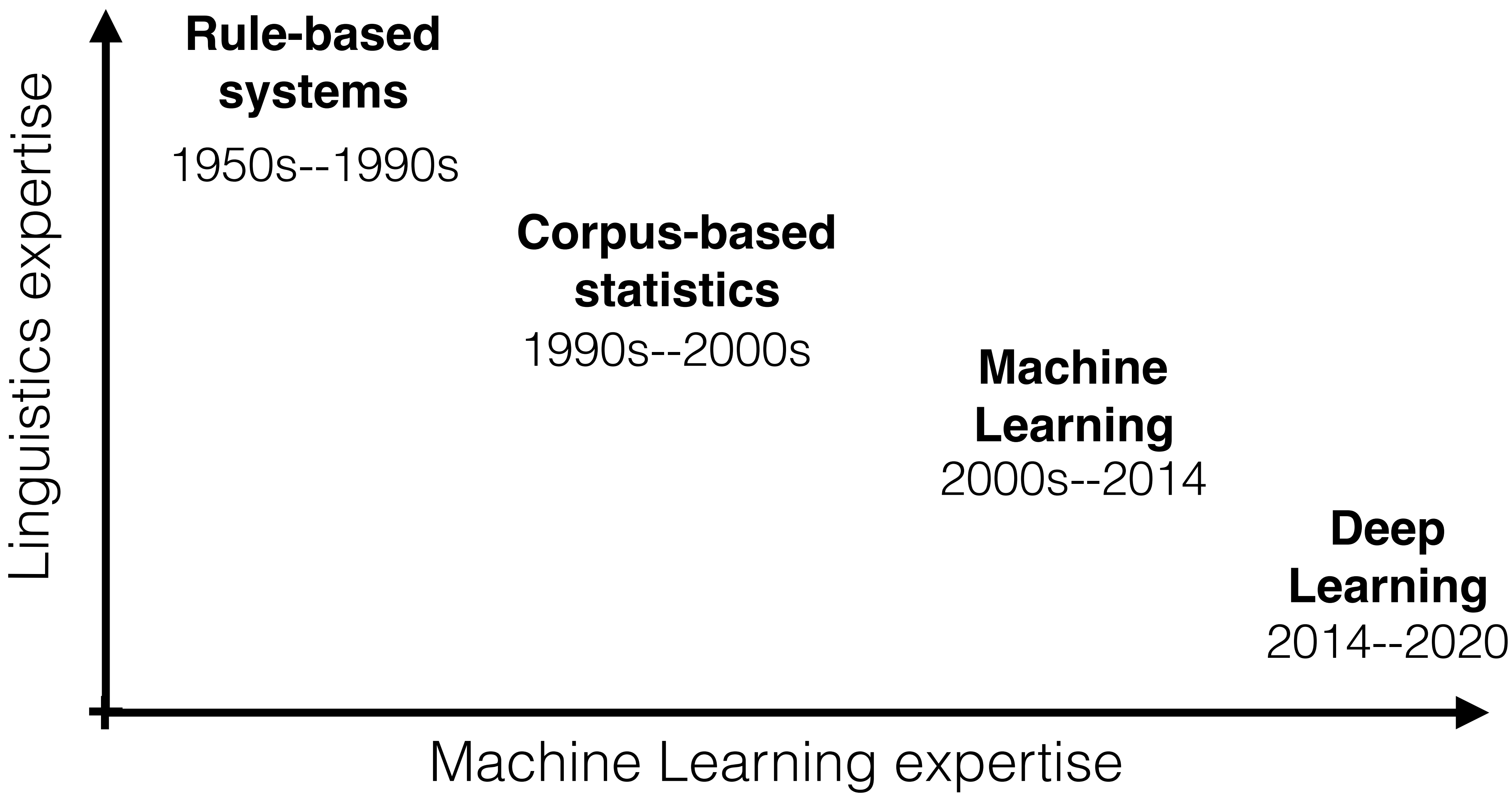


How do we do NLP?



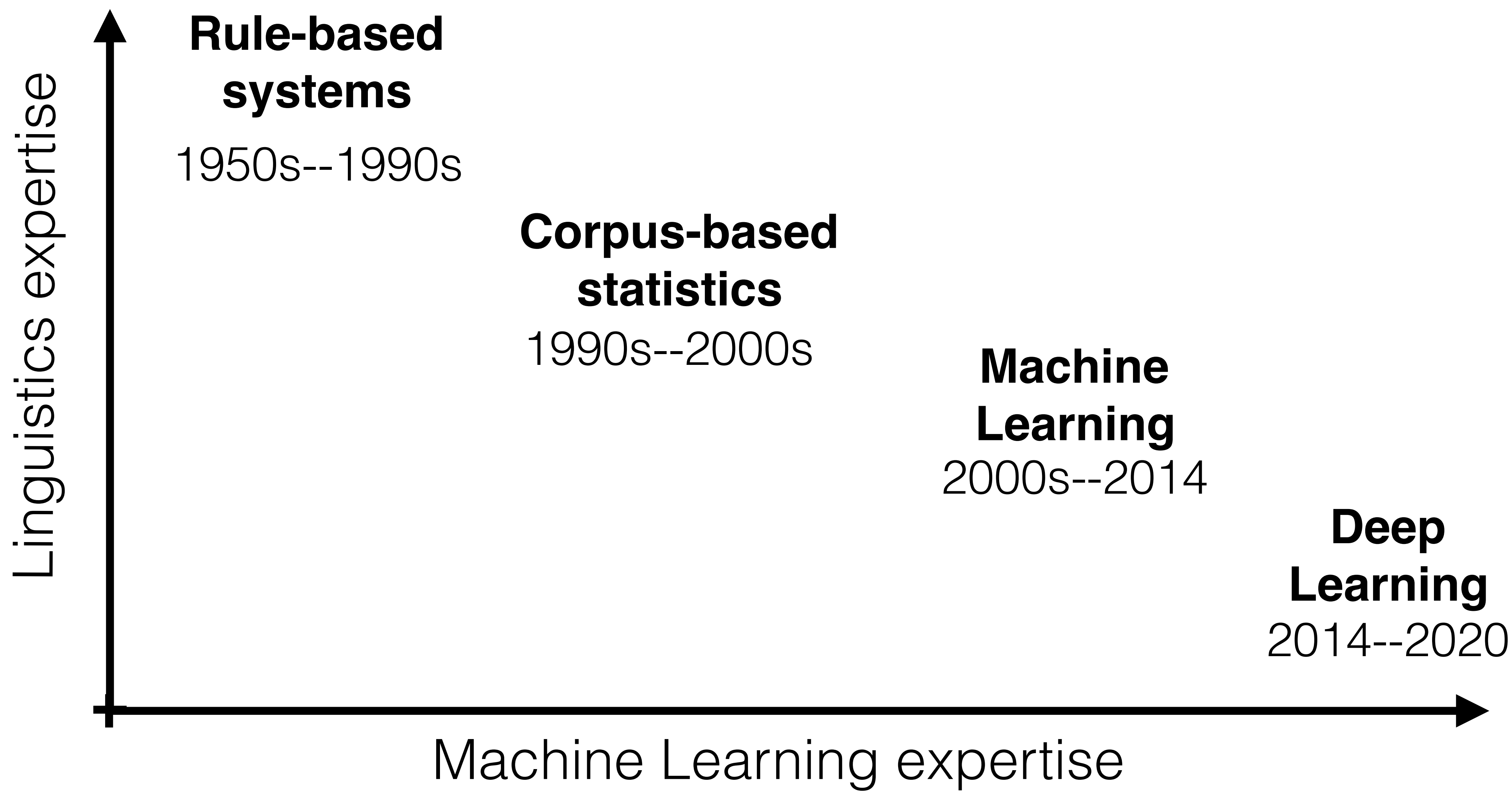
Transparent Debugability **Black box**

How do we do NLP?



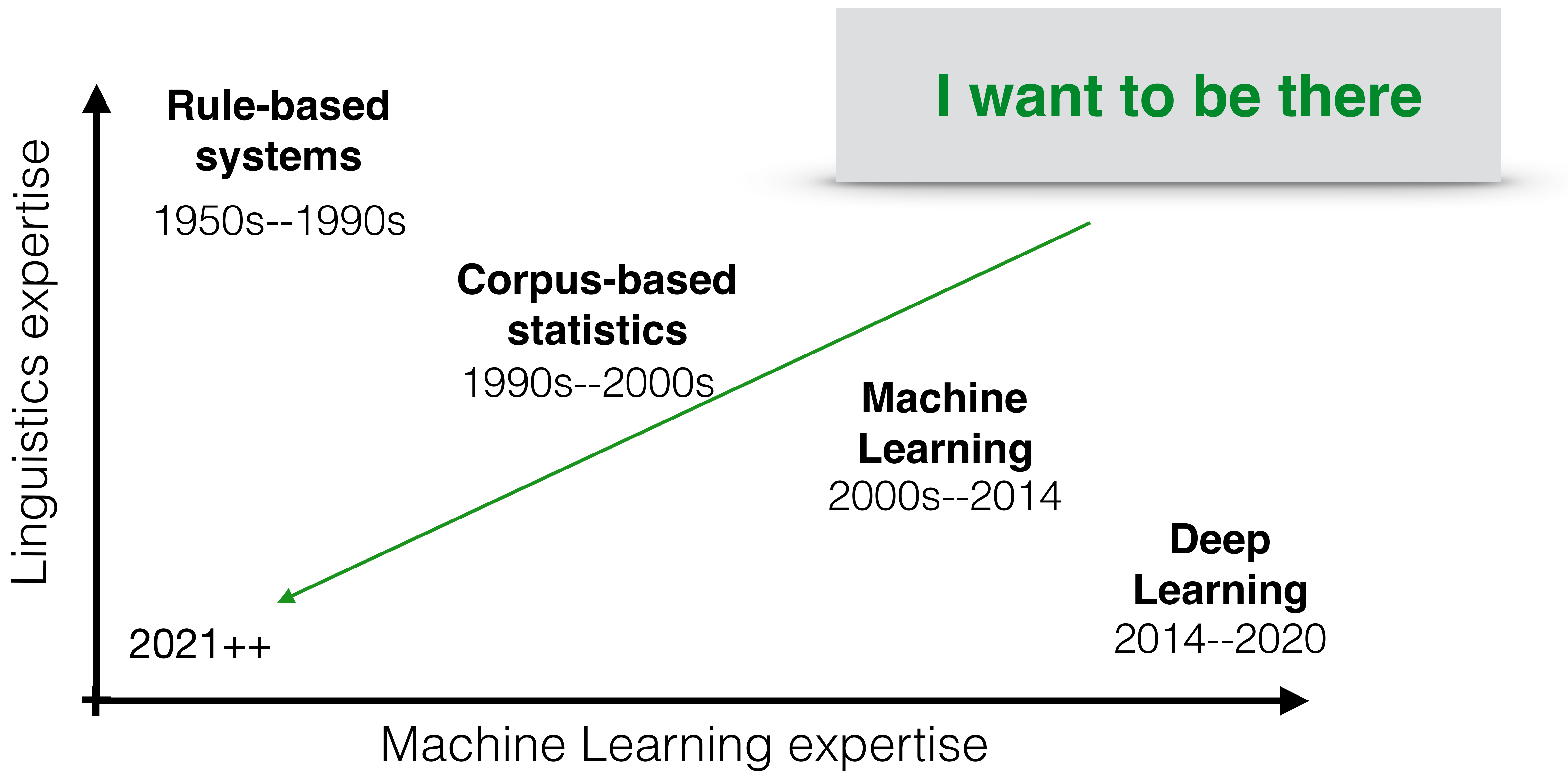
Transparent	Debugability	**Black box**
Little	Needed labeled data	**Lots**

How should we do NLP?



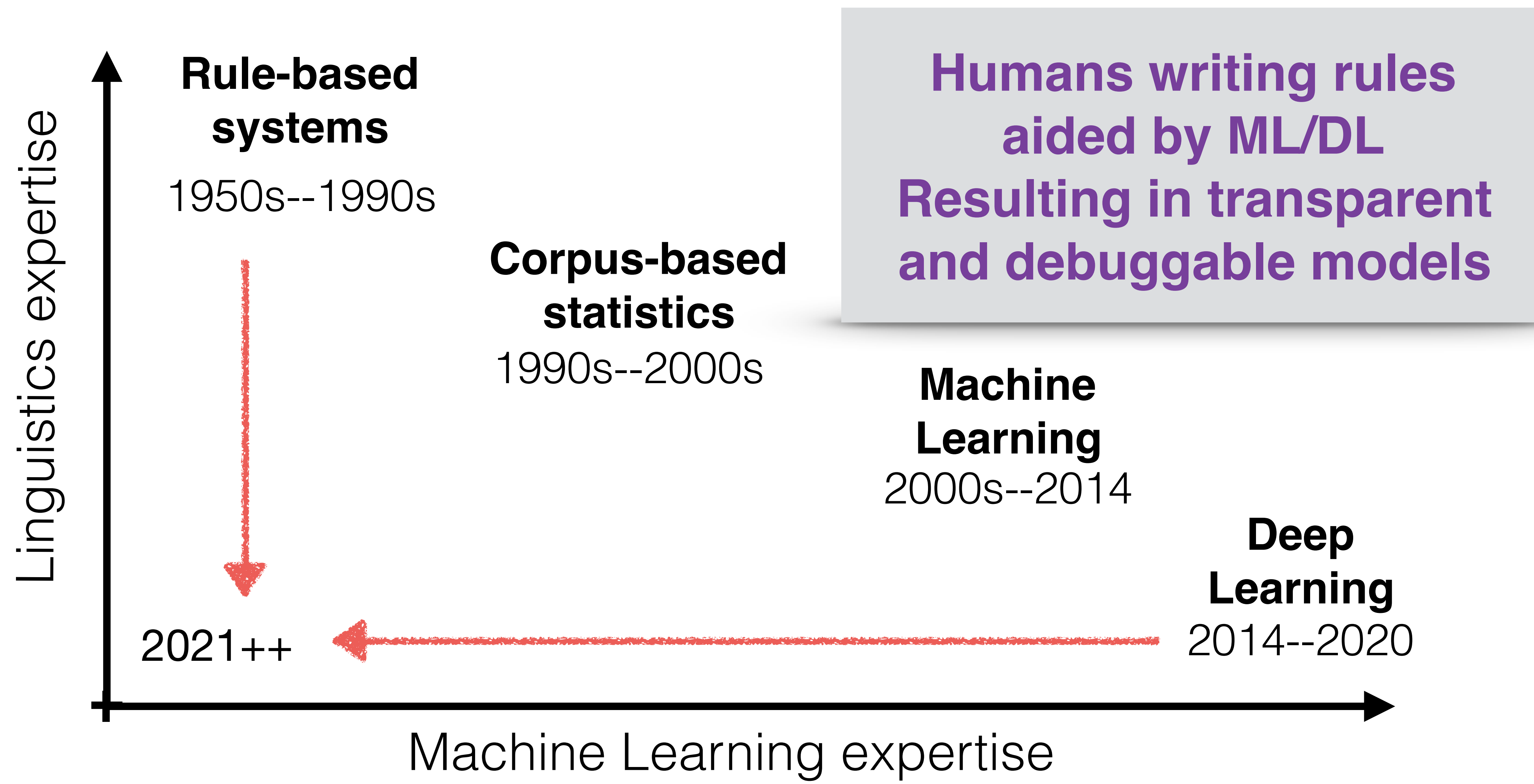
Transparent	Debugability	**Black box**
Little	Needed labeled data	**Lots**

How should we do NLP?



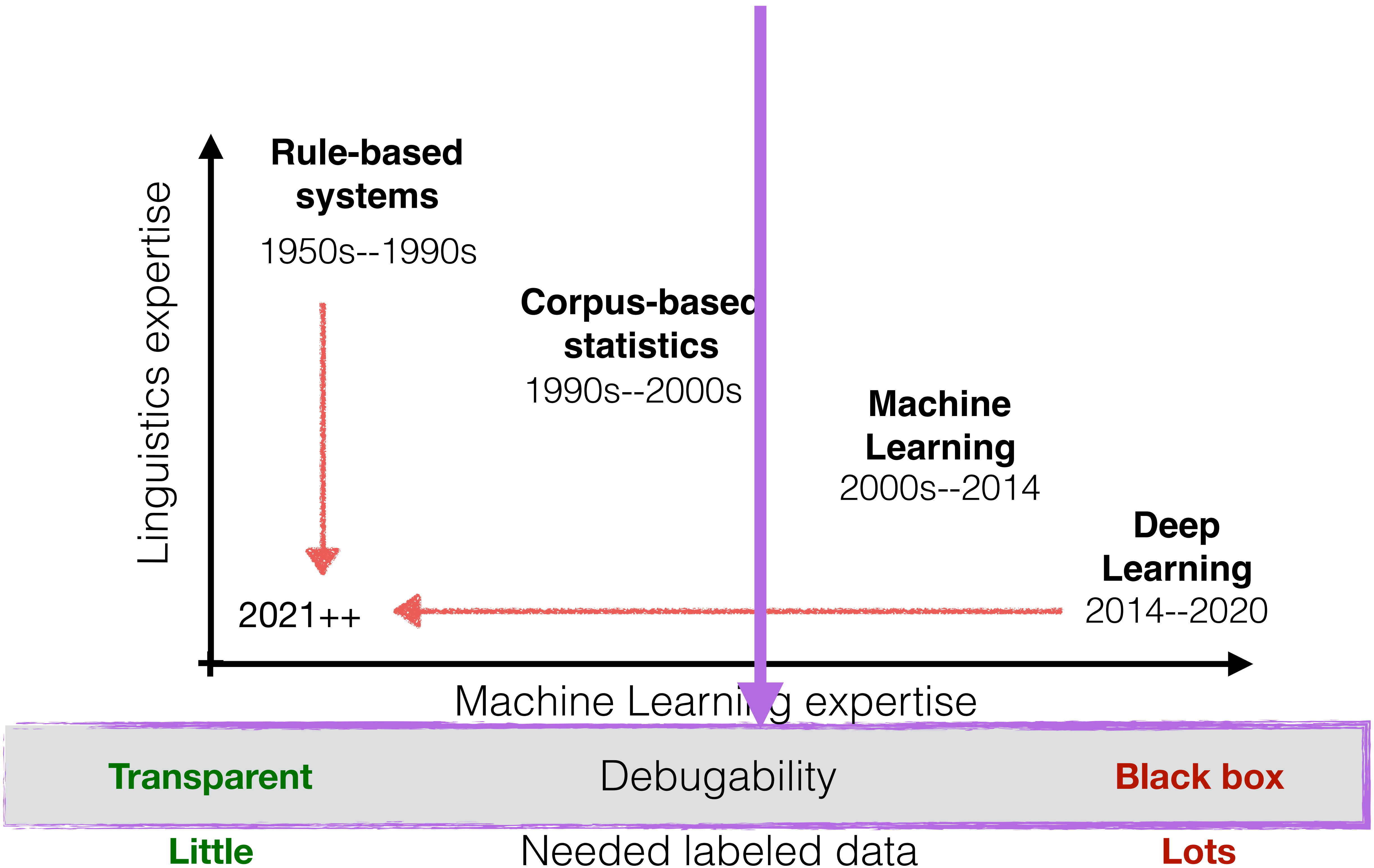
Transparent	Debugability	**Black box**
Little	Needed labeled data	**Lots**

NLP Tomorrow



Transparent	Debugability	**Black box**
Little	Needed labeled data	**Lots**

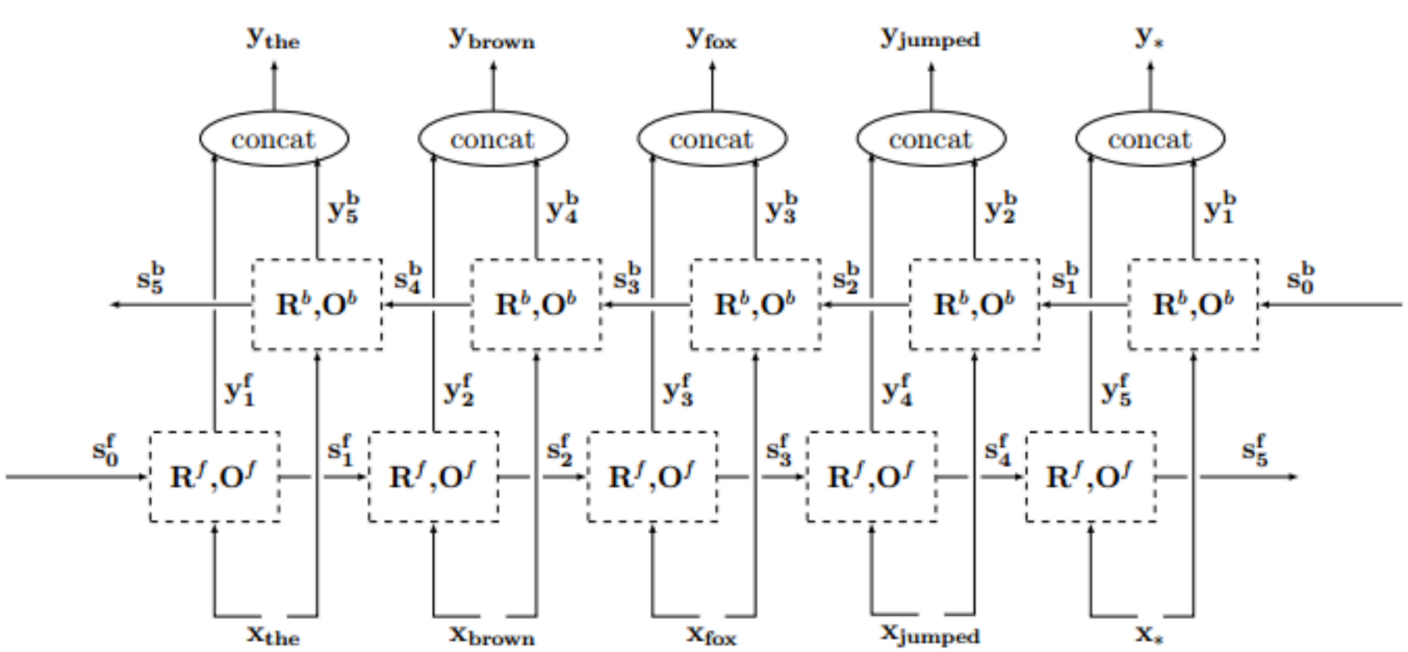
This lecture

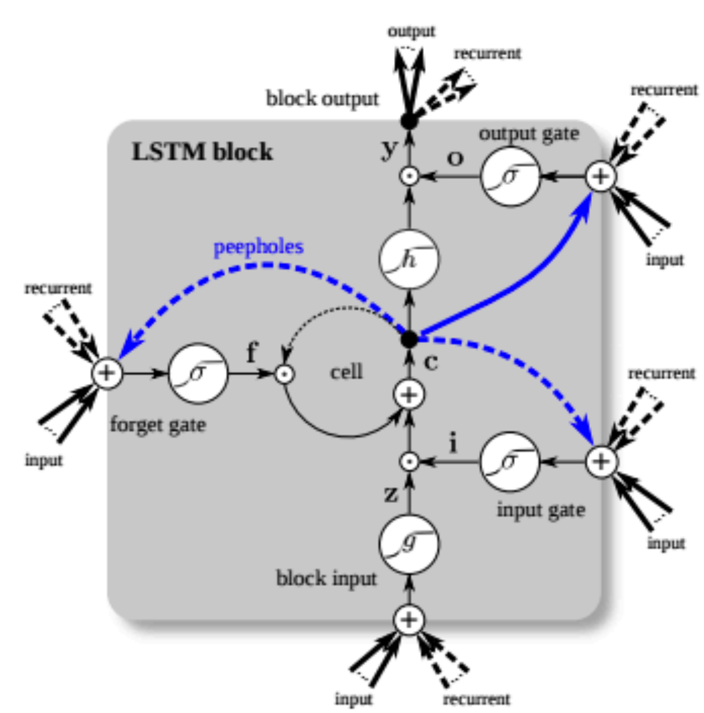


NLP Today

NLP Today

$$
\begin{aligned}
R_{LSTM}(\mathbf{s_{j-1}}, \mathbf{x_j}) &= [\mathbf{c_j}; \mathbf{h_j}] \\
\mathbf{c_j} &= \mathbf{c_{j-1}} \odot \mathbf{f} + \mathbf{g} \odot \mathbf{i} \\
\mathbf{h_j} &= \tanh(\mathbf{c_j}) \odot \mathbf{o} \\
\mathbf{i} &= \sigma(\mathbf{W^{xi}} \cdot \mathbf{x_j} + \mathbf{W^{hi}} \cdot \mathbf{h_{j-1}}) \\
\mathbf{f} &= \sigma(\mathbf{W^{xf}} \cdot \mathbf{x_j} + \mathbf{W^{hf}} \cdot \mathbf{h_{j-1}}) \\
\mathbf{o} &= \sigma(\mathbf{W^{xo}} \cdot \mathbf{x_j} + \mathbf{W^{ho}} \cdot \mathbf{h_{j-1}}) \\
\mathbf{g} &= \tanh(\mathbf{W^{xg}} \cdot \mathbf{x_j} + \mathbf{W^{hg}} \cdot \mathbf{h_{j-1}})
\end{aligned}
$$





NLP Today

3. The BiLSTM Hegemony

**To a first approximation,
the de facto consensus in NLP in 2017 is
that no matter what the task,
you throw a BiLSTM at it, with
attention if you need information flow**

28

Chris Manning
April 2017

NLP Today

3. The BiLSTM Hegemony

**To a first approximation,
the de facto consensus in NLP in 2017 is
that no matter what the task,
you throw a BiLSTM at it, with
attention if you need information flow**

28

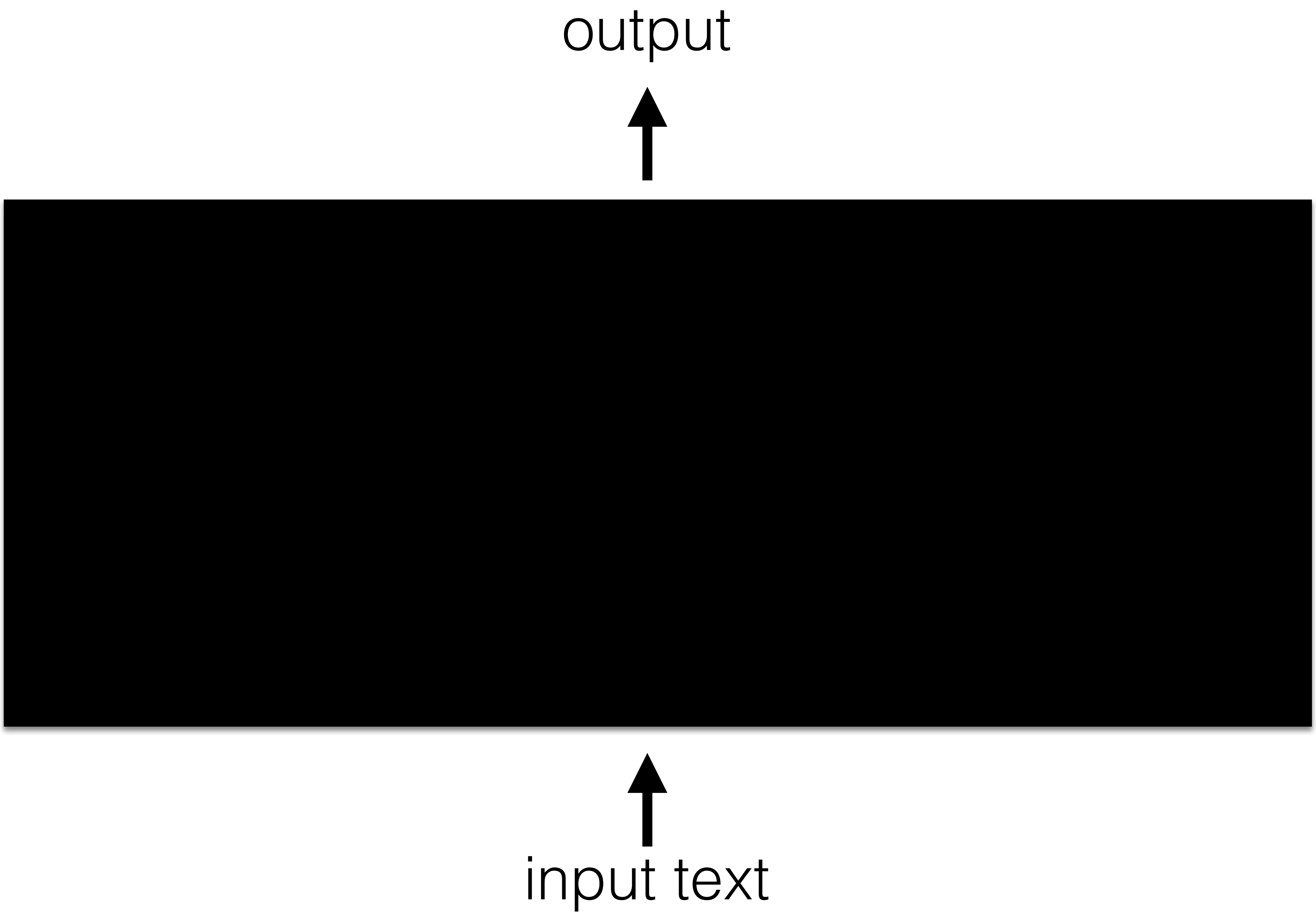
Chris Manning
April 2017

BIU NLP

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

NLP Today

NLP Today

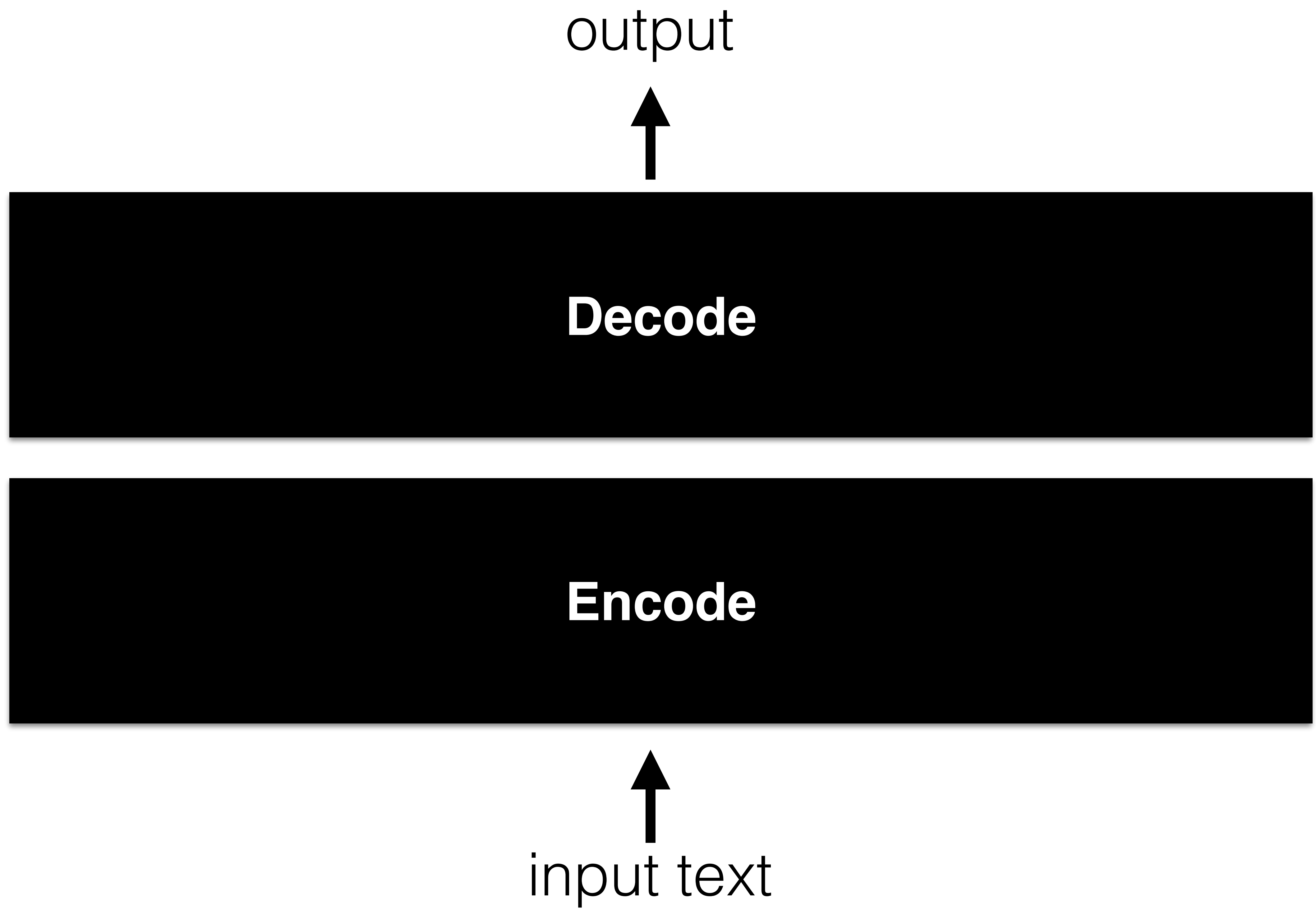


NLP Today

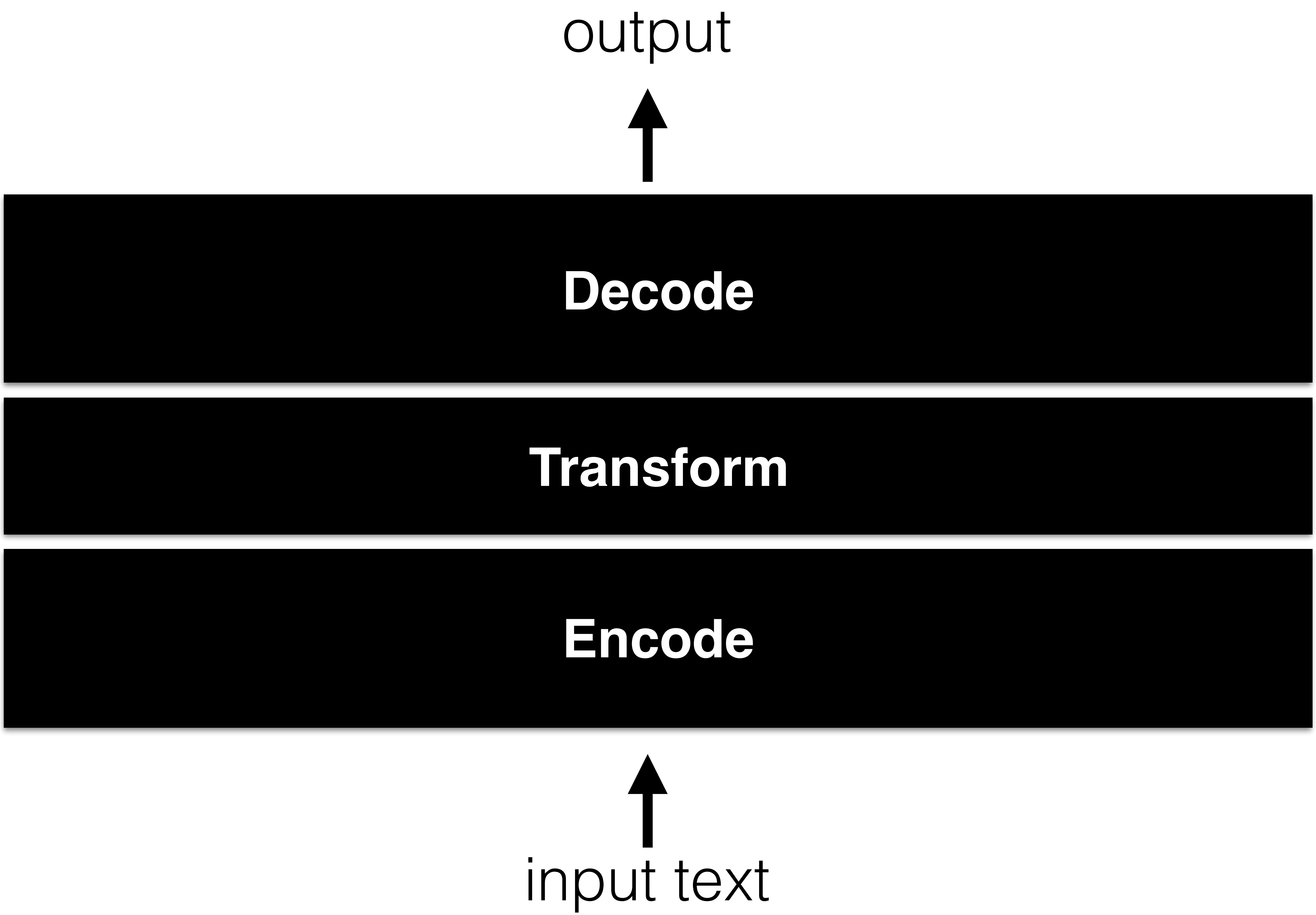
output

input text

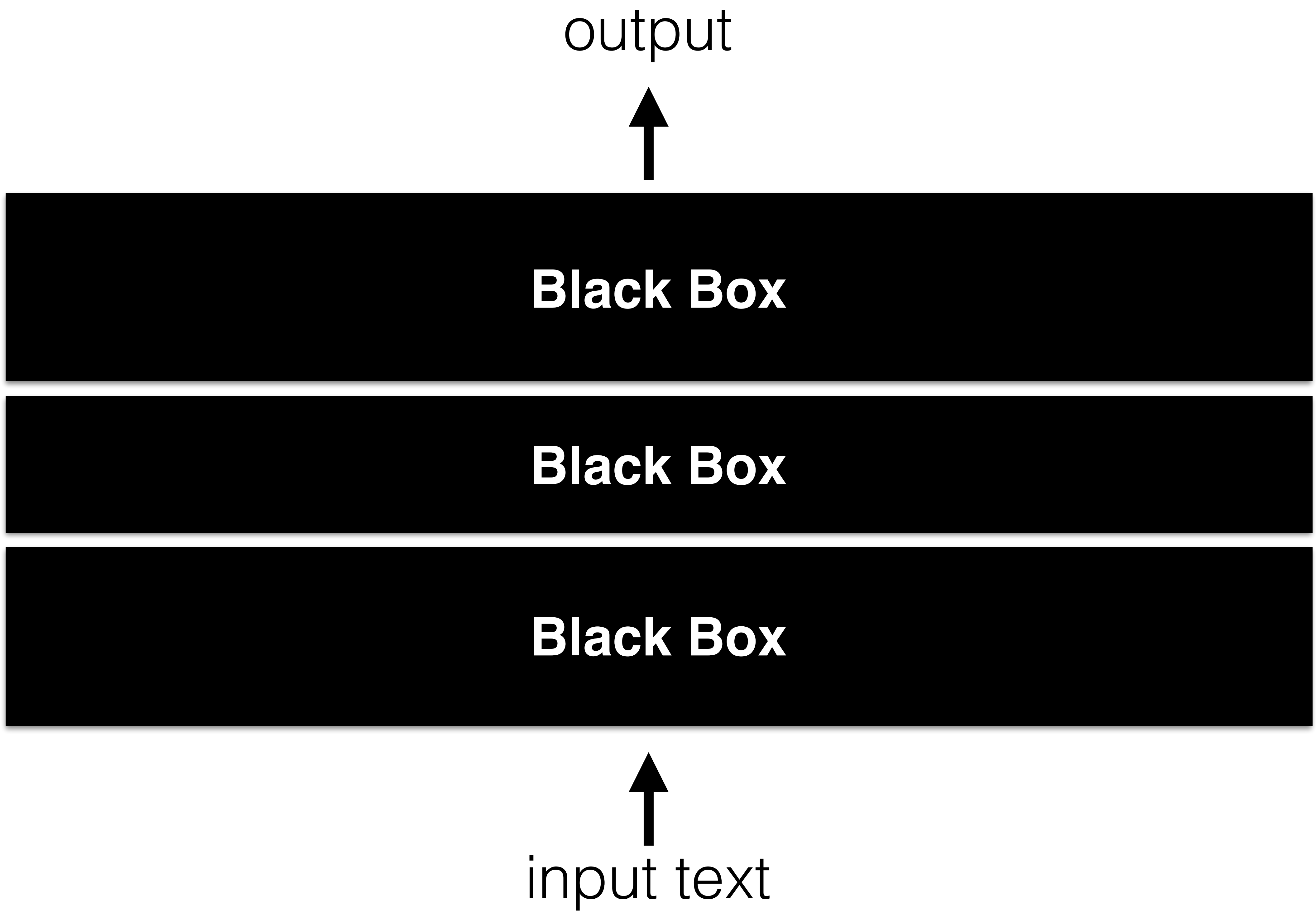
NLP Today



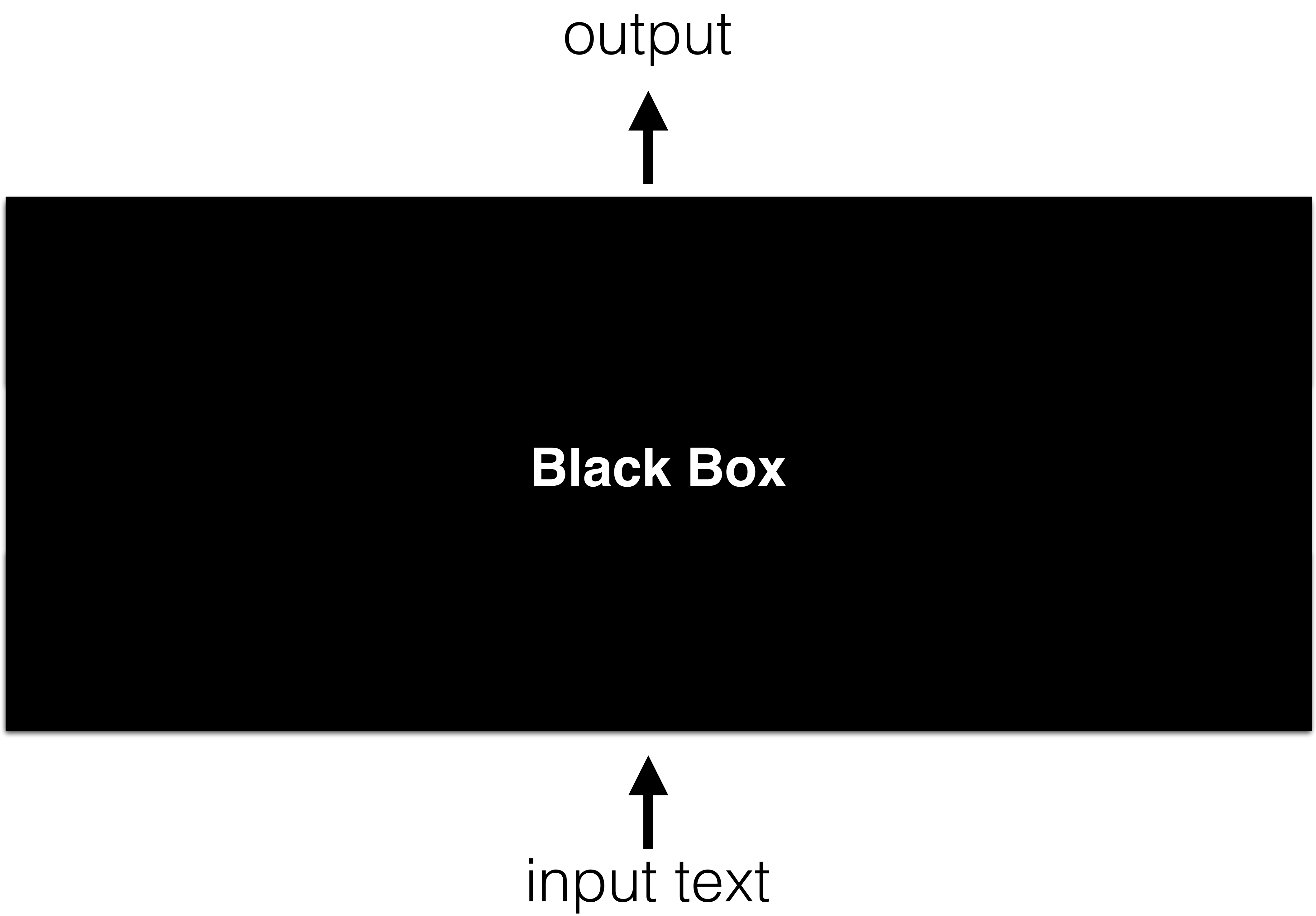
NLP Today



NLP Today



NLP Today



Black Boxes

- How do these black boxes work?
- What **can** they learn / represent?
- What **did** they learn / represent?

Black Boxes

Analyzing and interpreting neural networks for NLP

Revealing the content

BlackboxNLP 2019

The second edition of the BlackboxNLP workshop will be collocated with ACL 2019 in Florence.

Archived information about the 2018 edition: blackboxnlp.github.io/2018.

Important dates

- ~~April 19~~. Submission deadline (11:59pm Pacific Daylight Savings Time, UTC-7h).
- ~~May 17~~ ~~**May 20**~~. Notification of acceptance.
- ~~June 3~~. Camera ready (11:59pm Pacific Daylight Savings Time, UTC-7h).
- August 1. Workshop.

Many research Qs

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what did they *really* learn?

Q5: What is the representation power of diff archs?

Q6: Extracting a discrete reps from a trained model.

Many research Qs

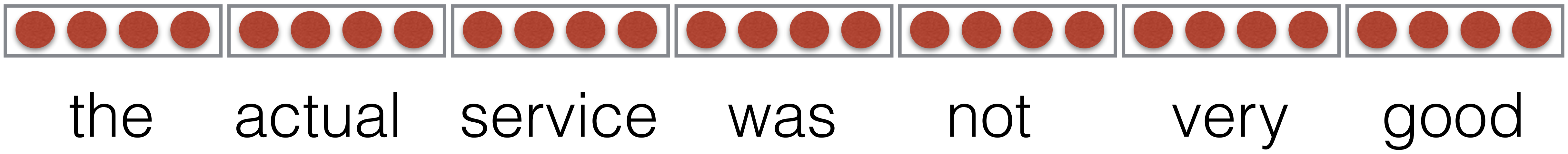
- How do these black boxes work?
- What **can** they learn / represent?
- What **did** they learn / represent?

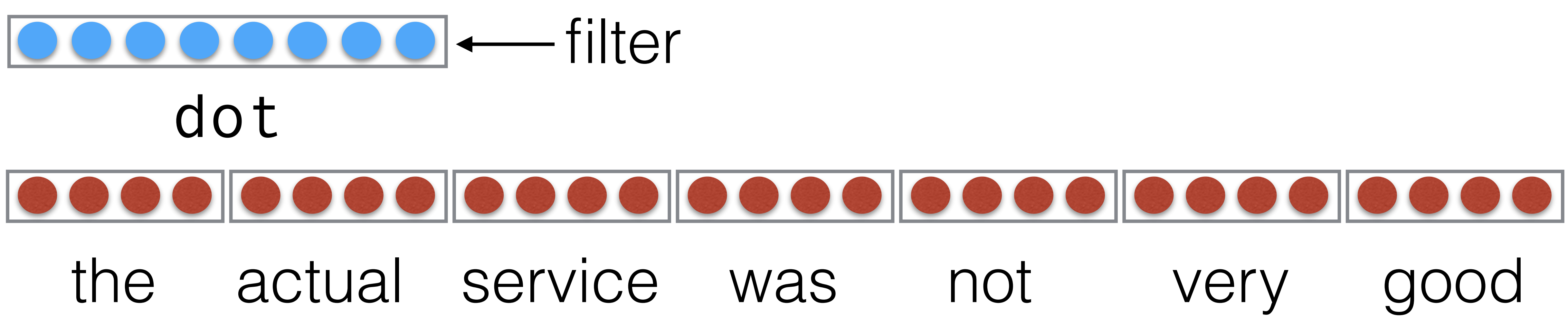
Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

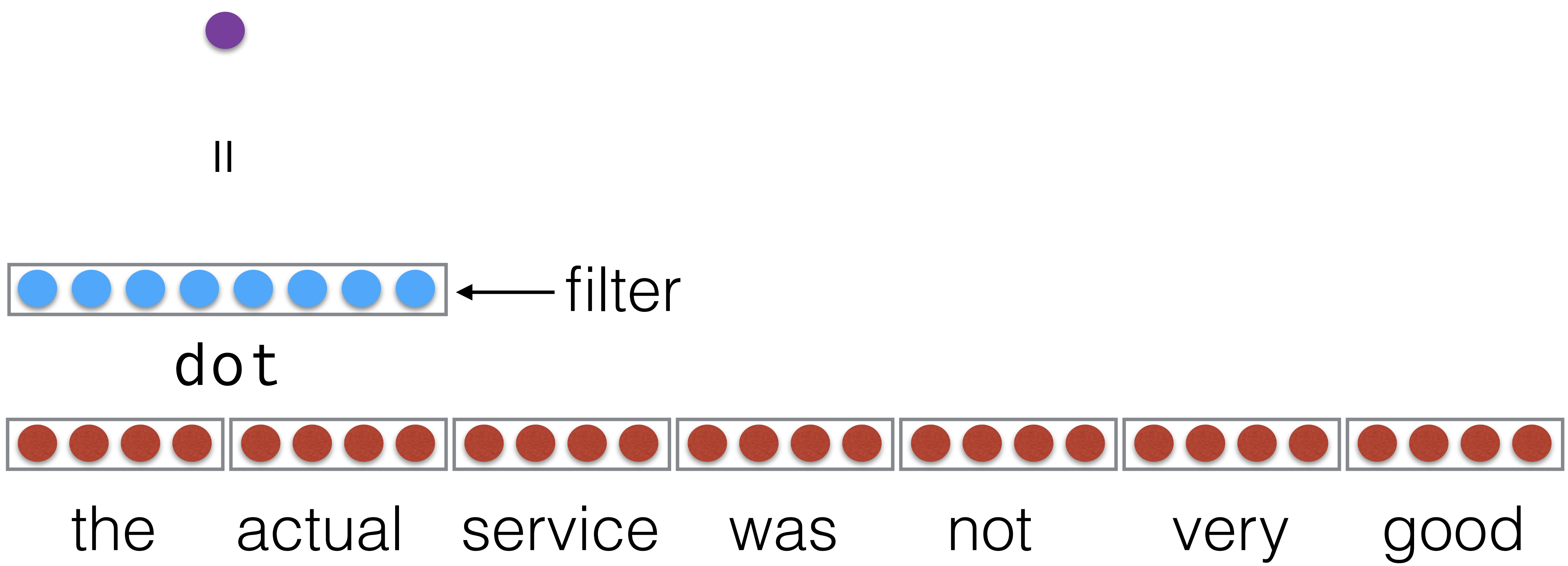
The learned functions are complex.

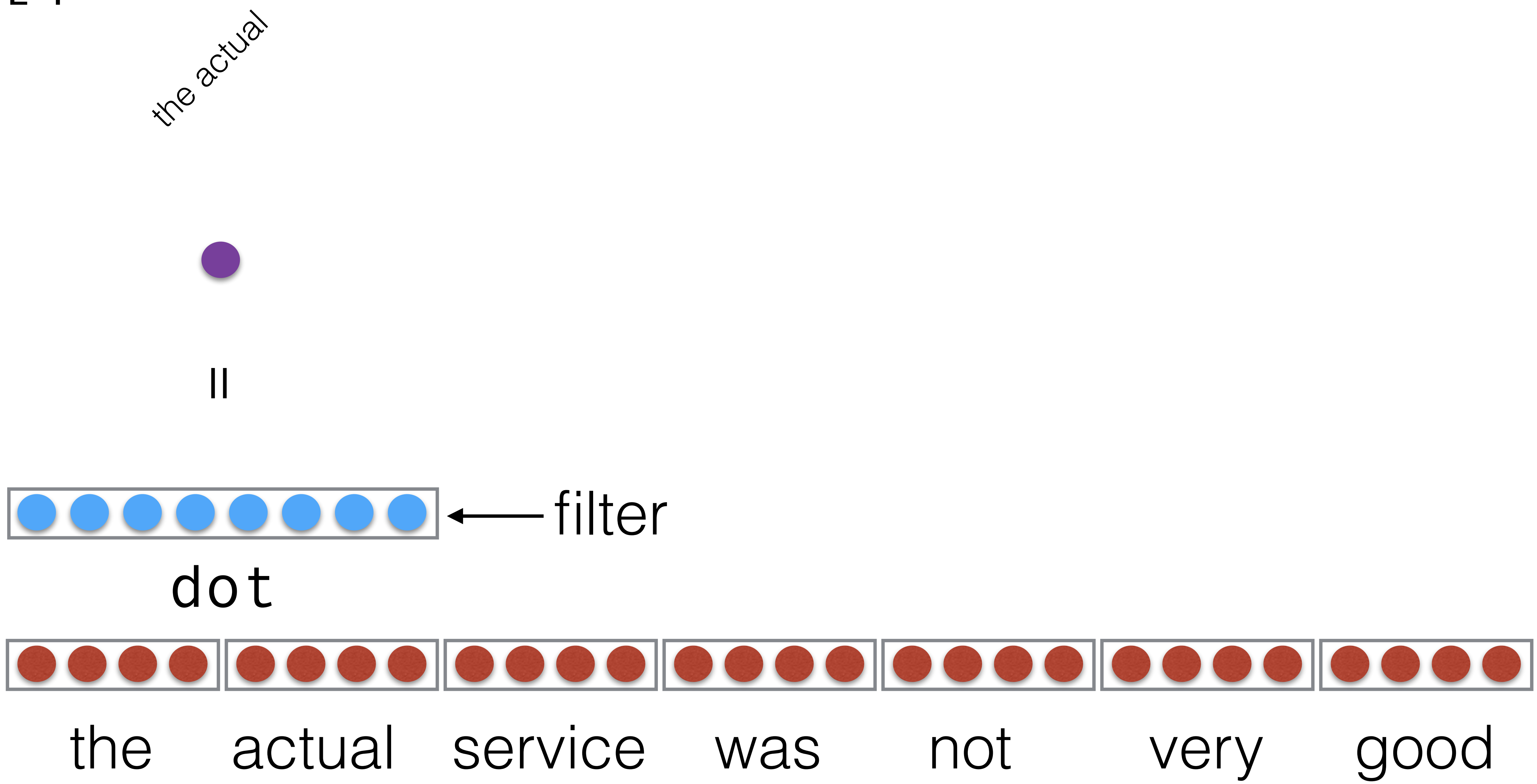
Our intuitions are often wrong.

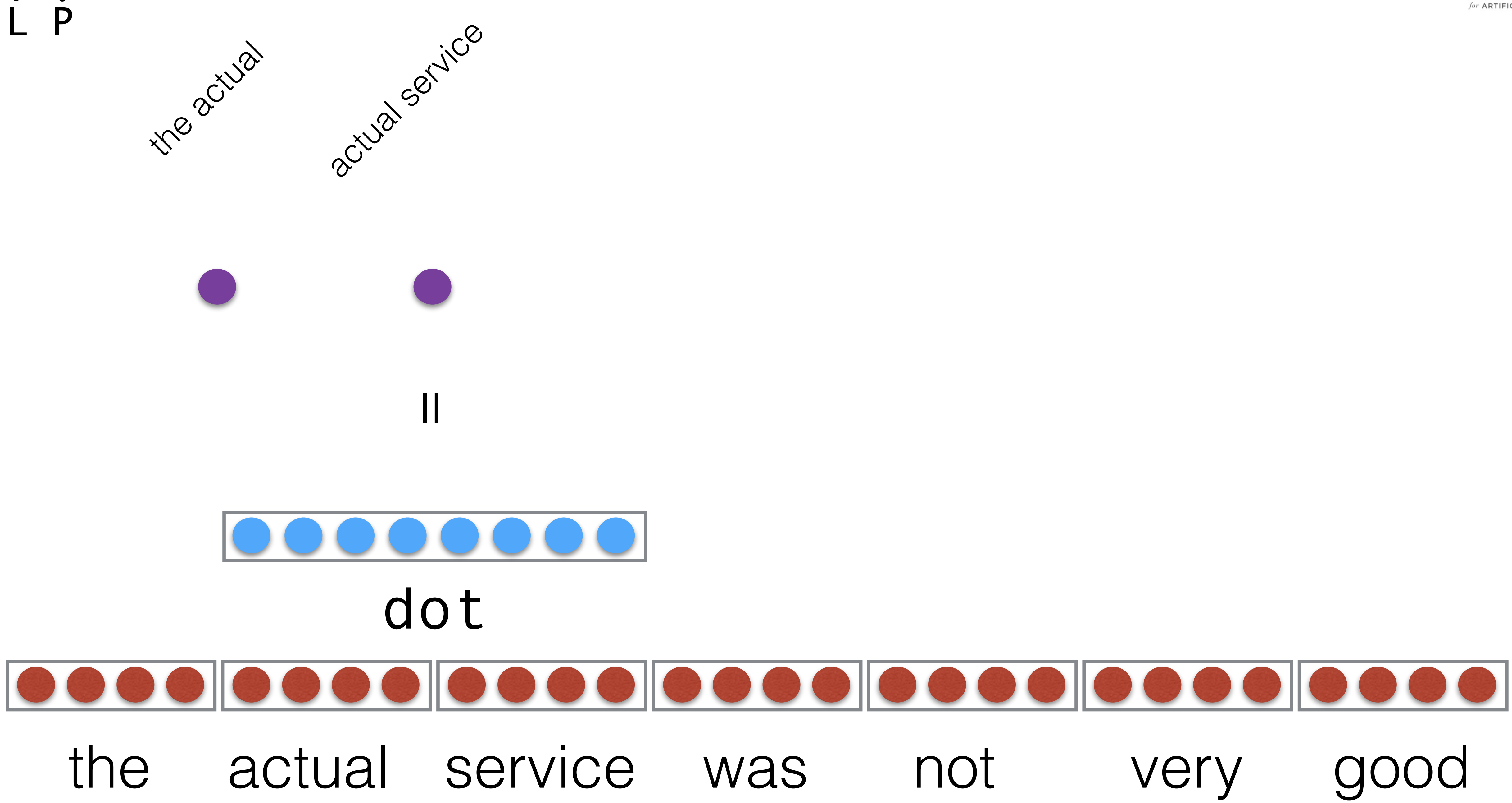
Intro to 1D CNN

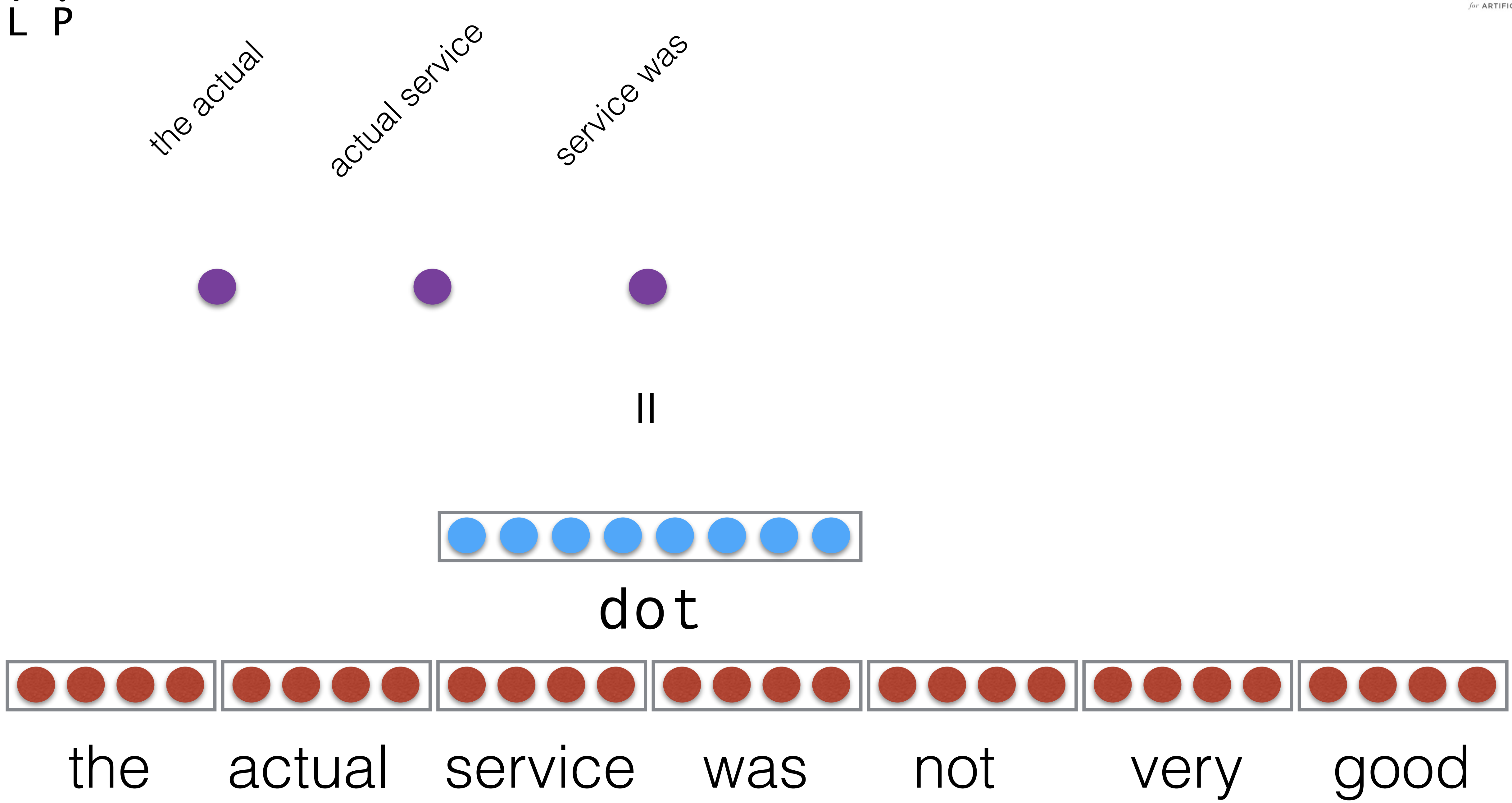

the
actual
service
was
not
very
good


filter
dot
the
actual
service
was
not
very
good

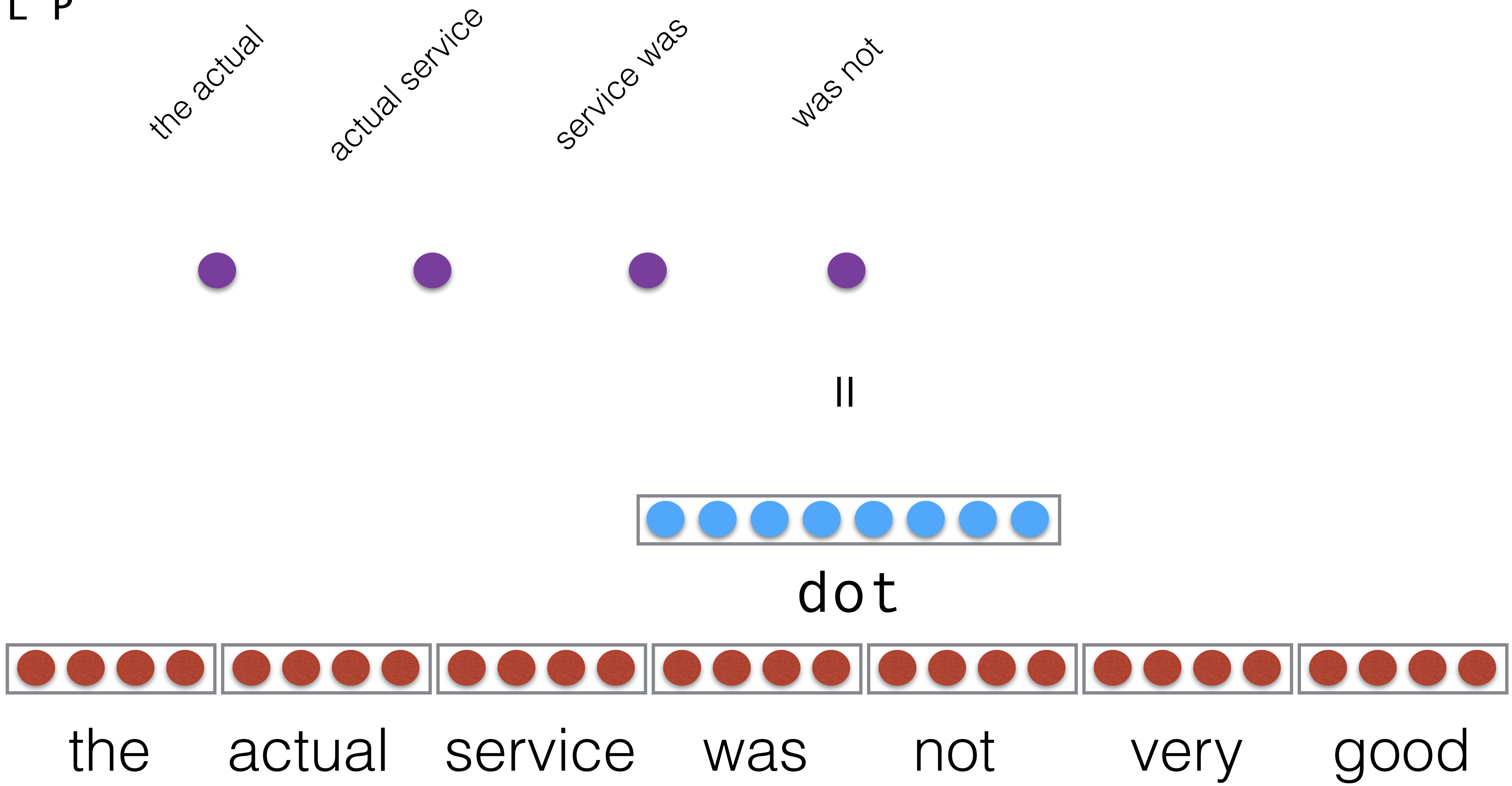

=
filter
dot
the
actual
service
was
not
very
good

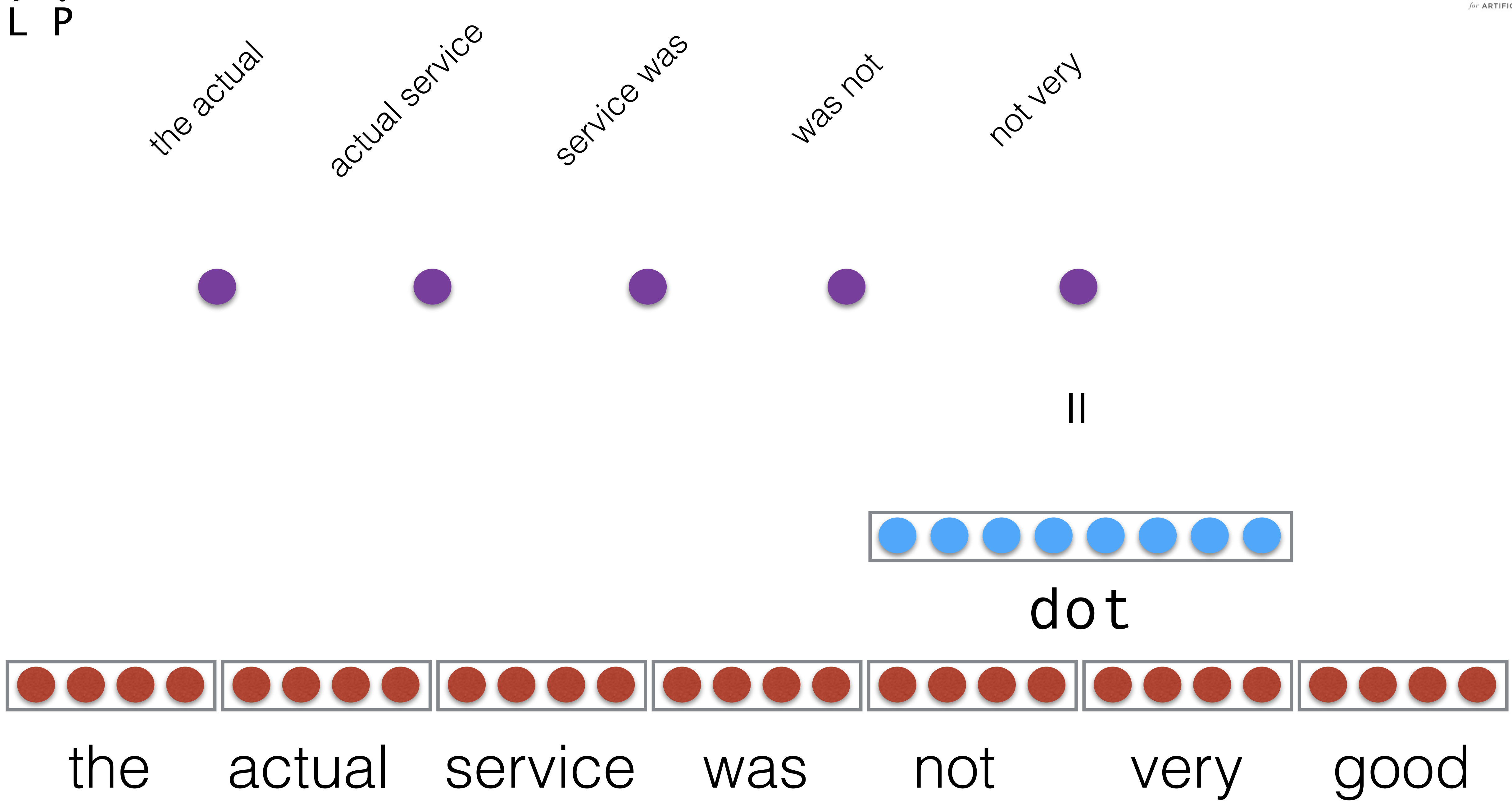

the actual
=
filter
dot
the
actual
service
was
not
very
good


the actual
actual service
=
dot
the
actual
service
was
not
very
good

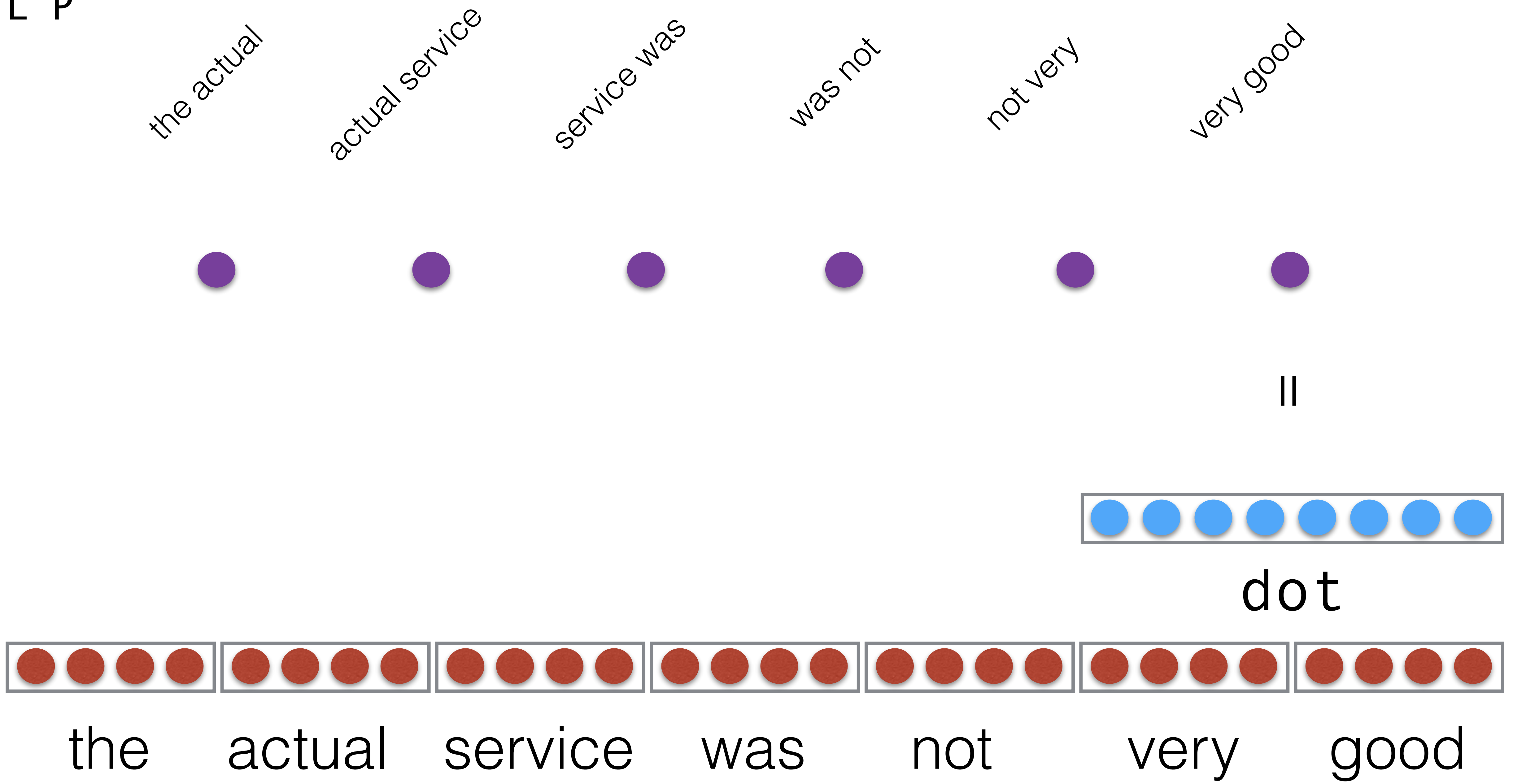


the actual
actual service
service was
=
dot
the
actual
service
was
not
very
good

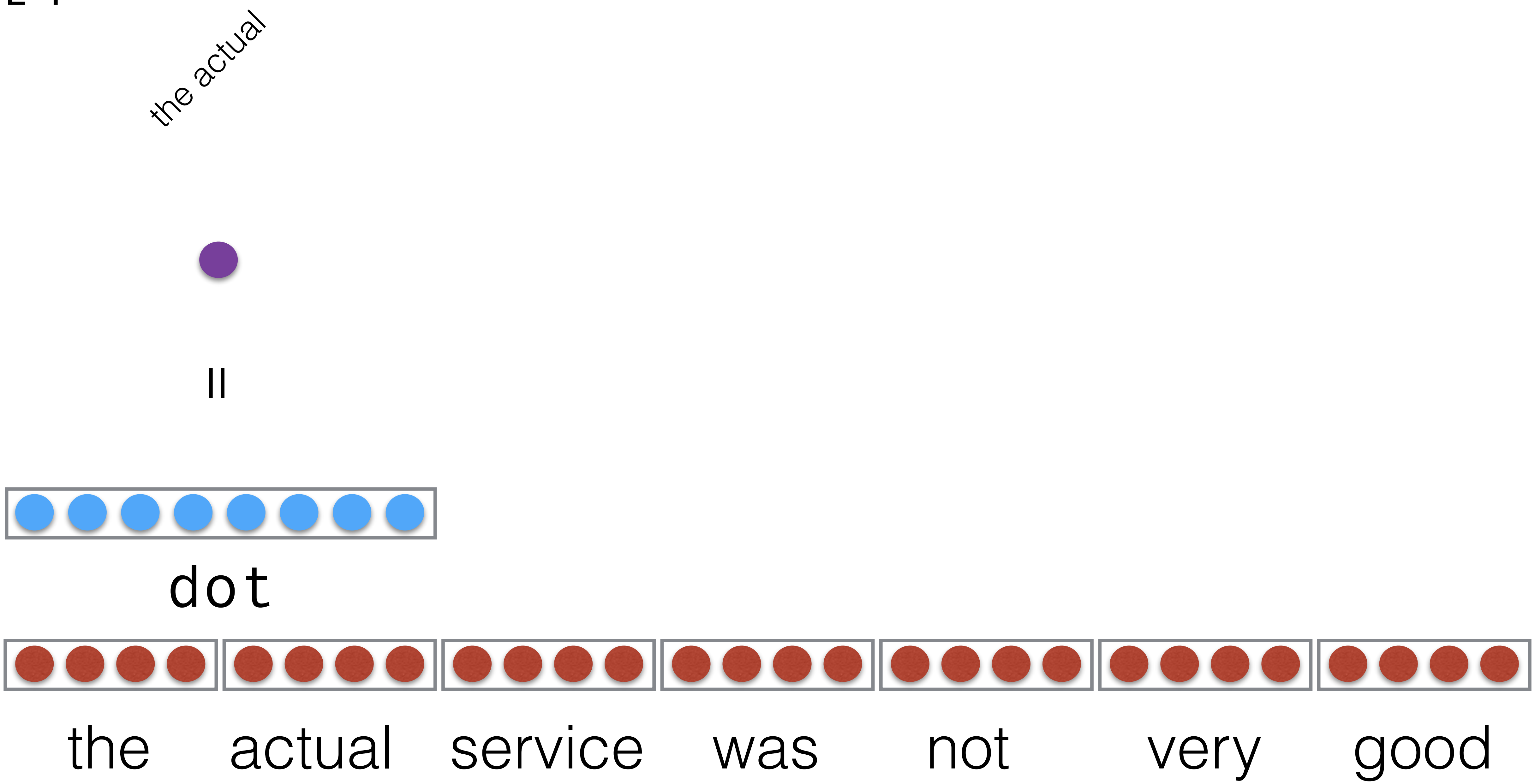

the actual
actual service
service was
was not
=
dot
the
actual
service
was
not
very
good

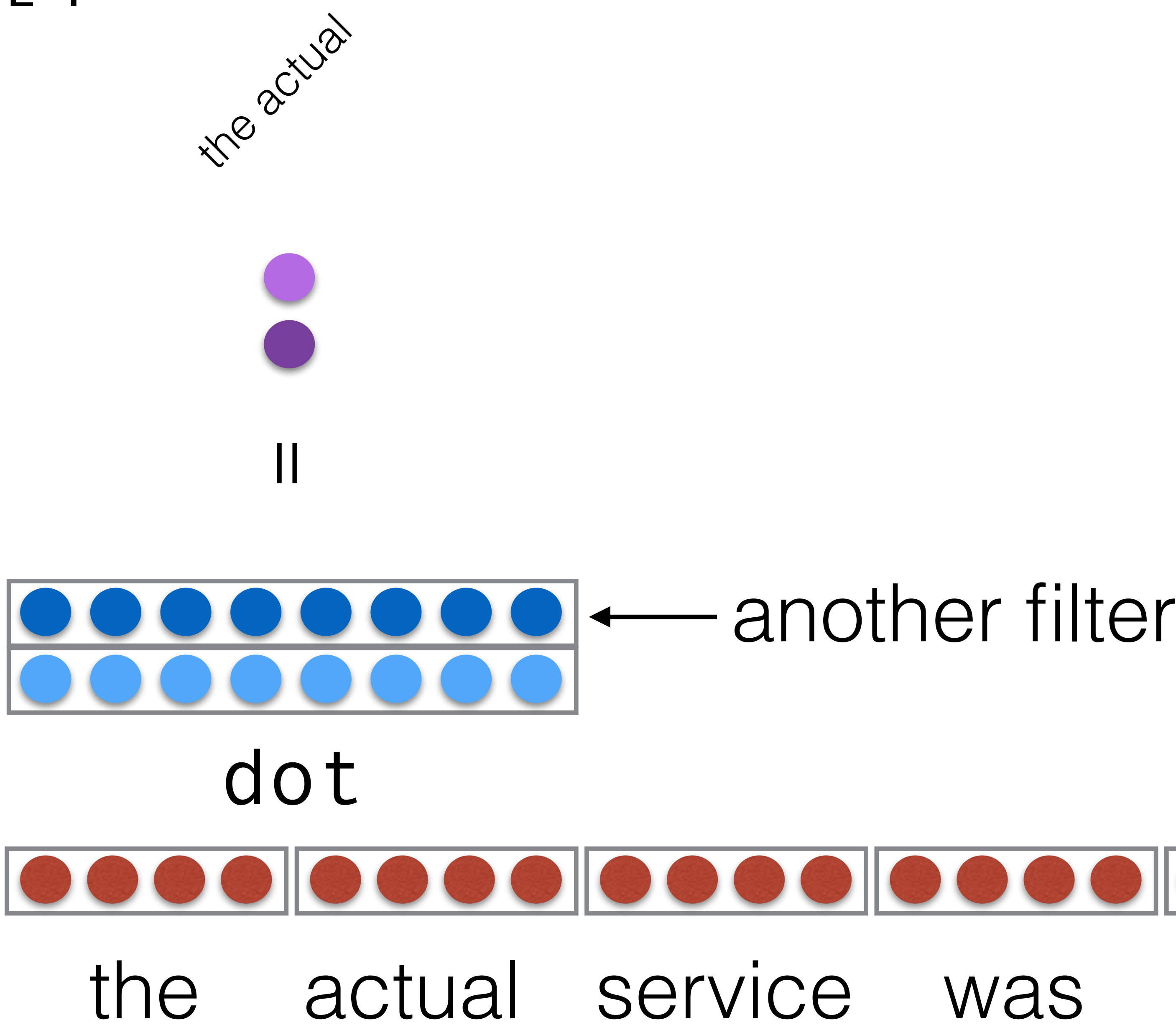


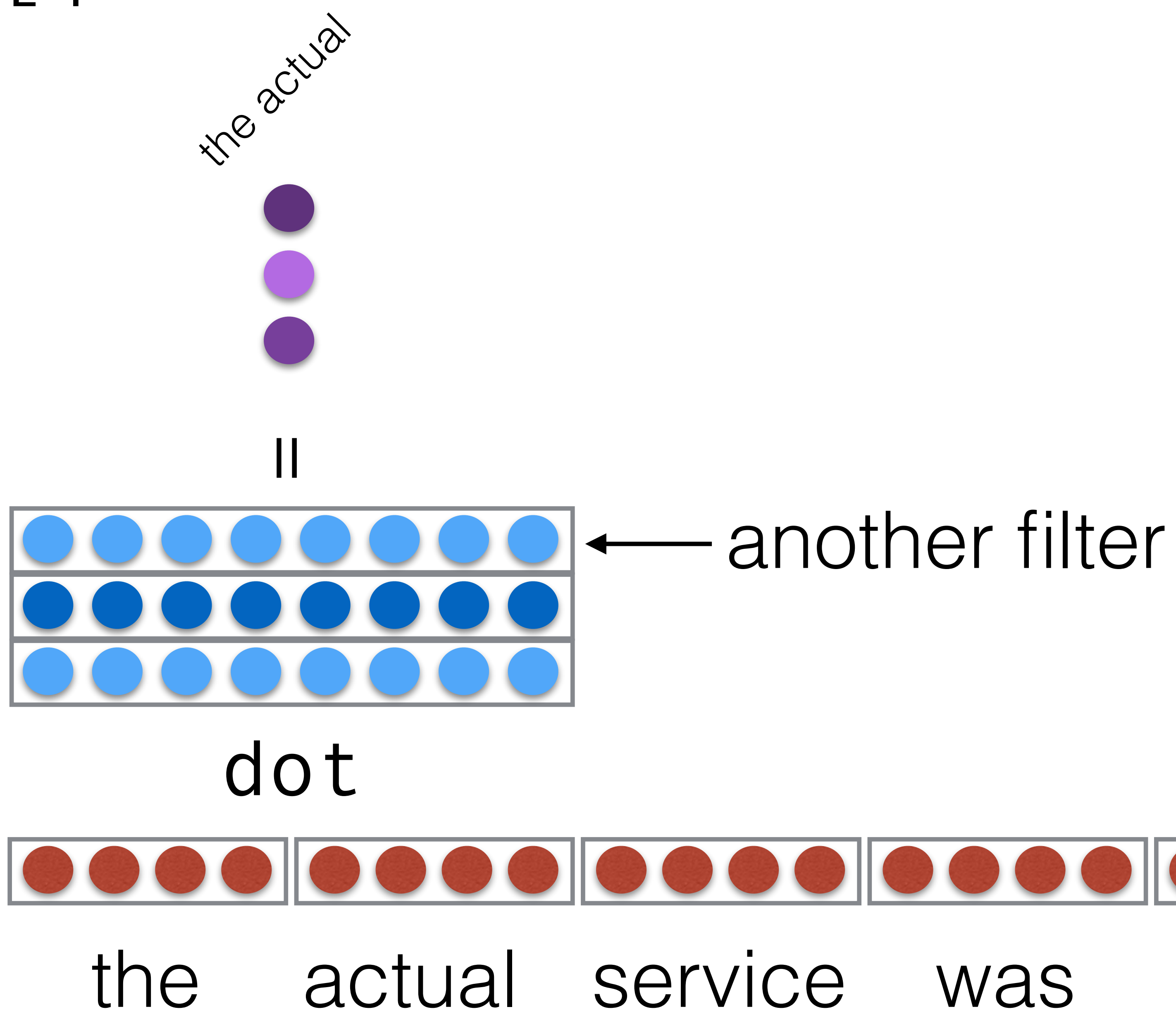
the actual
actual service
service was
was not
not very
=
dot
the
actual
service
was
not
very
good

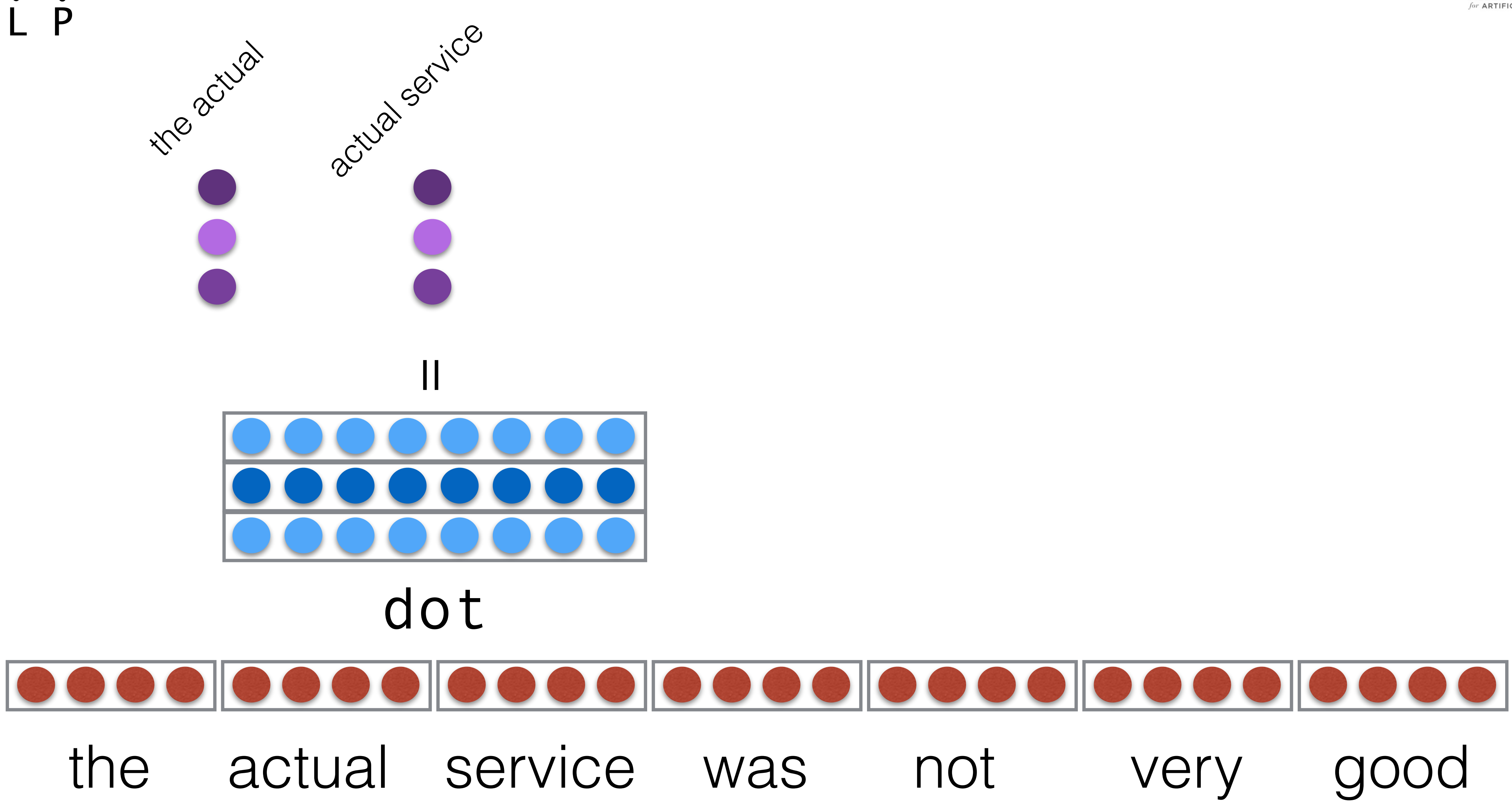


the actual
actual service
service was
was not
not very
very good
=
dot
the
actual
service
was
not
very
good

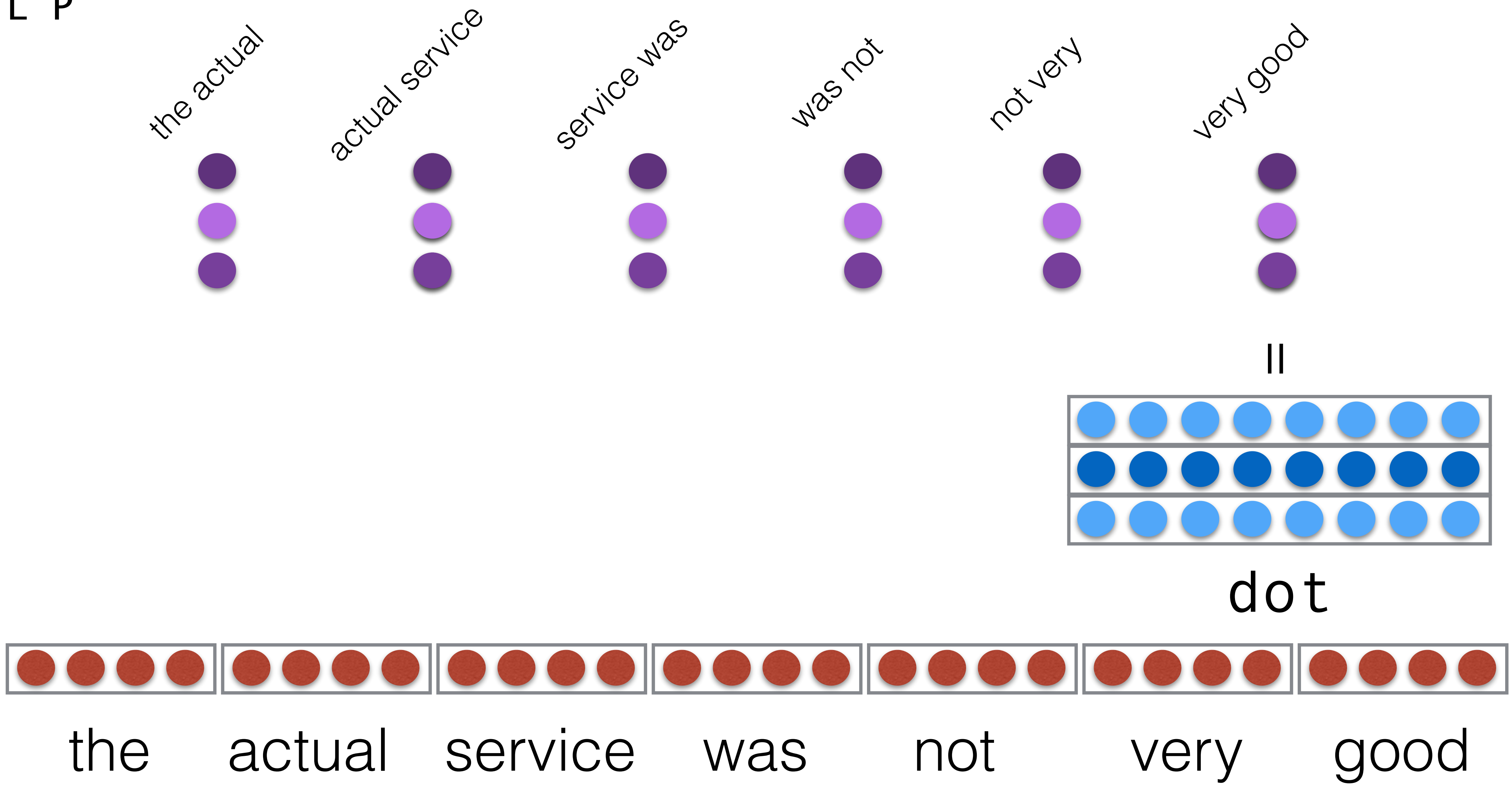

the actual
=
dot
the
actual
service
was
not
very
good


the actual
=
another filter
dot
the
actual
service
was
not
very
good

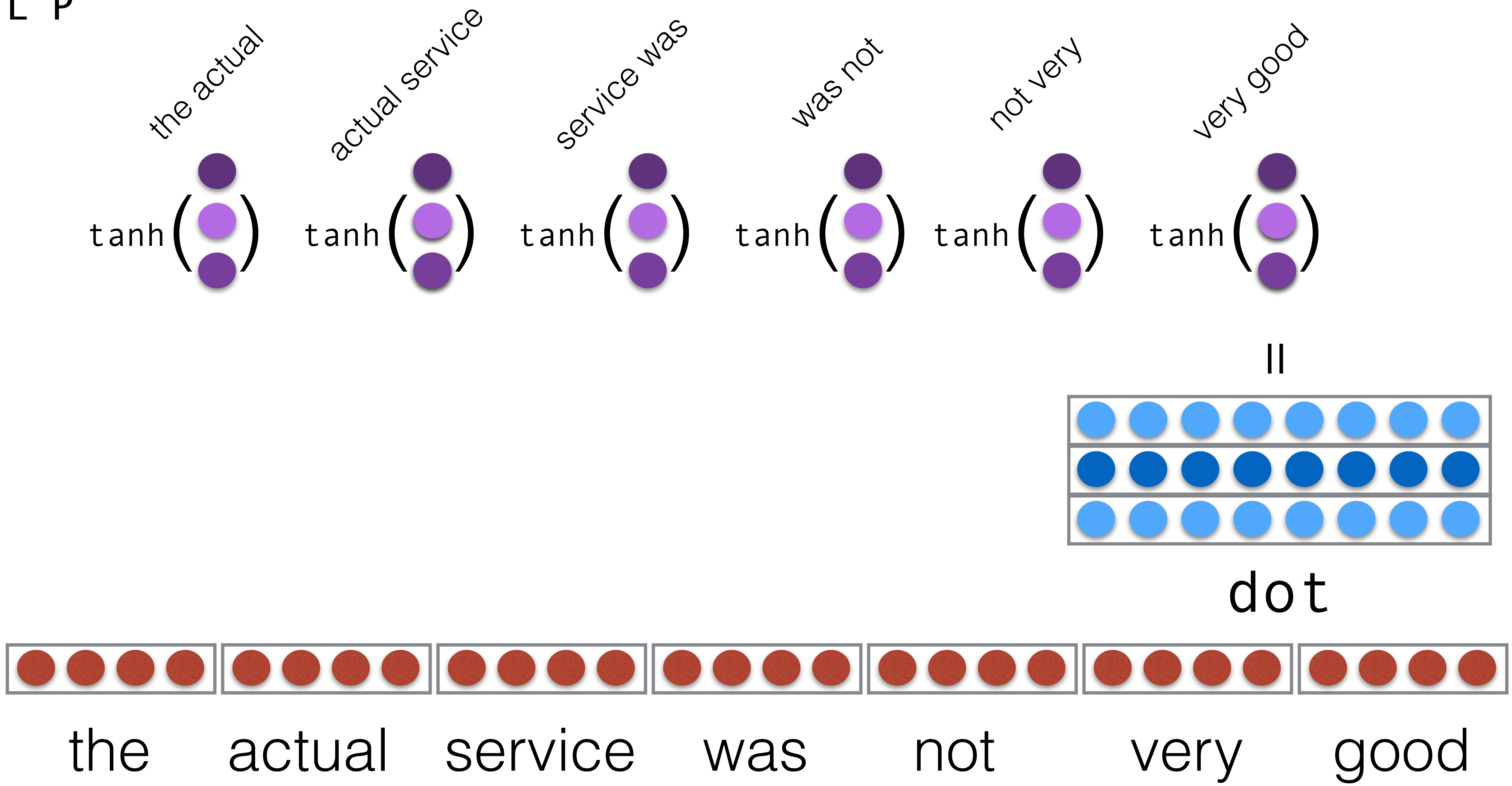

the actual
=
another filter
dot
the
actual
service
was
not
very
good



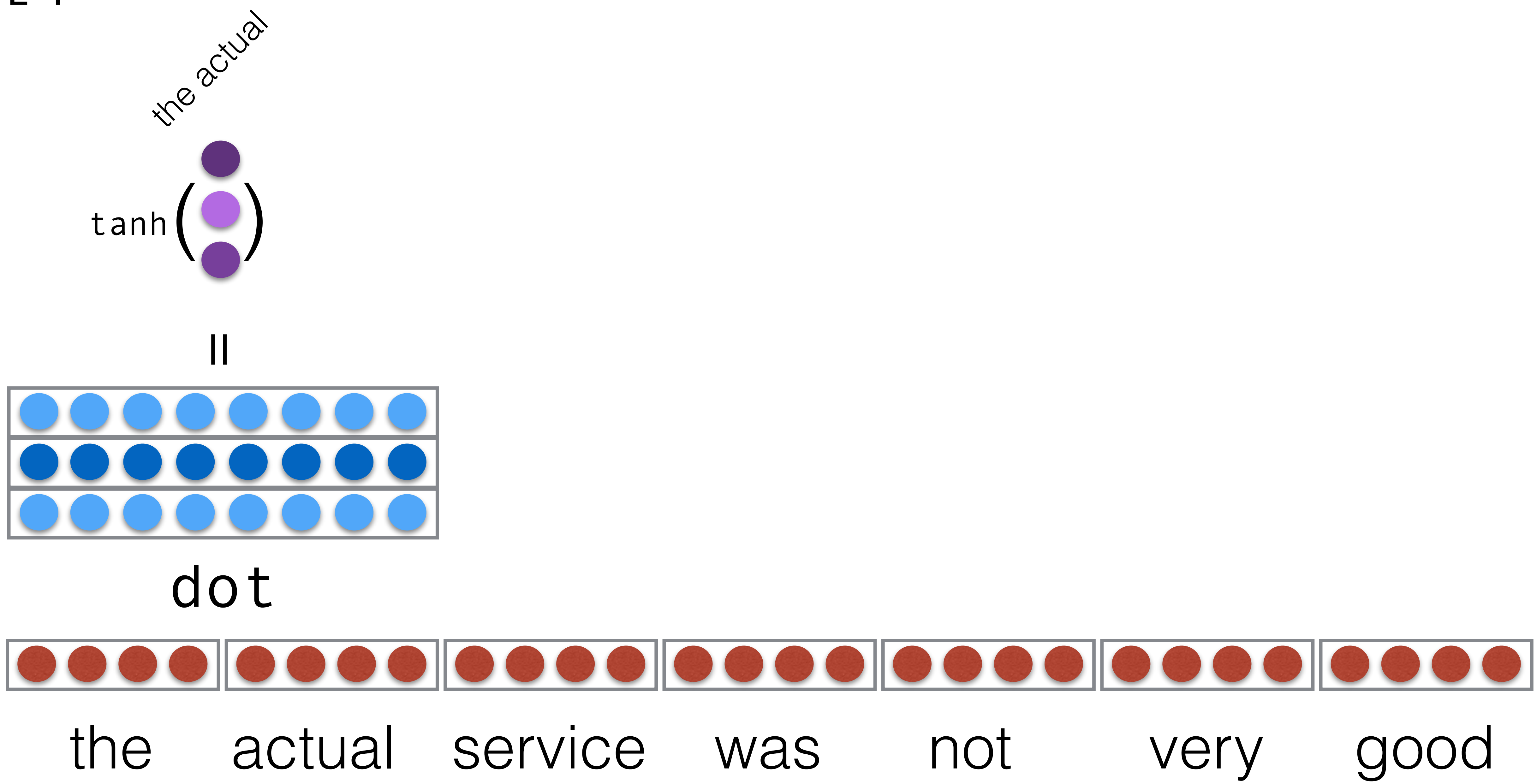
the actual
actual service
=
dot
the
actual
service
was
not
very
good



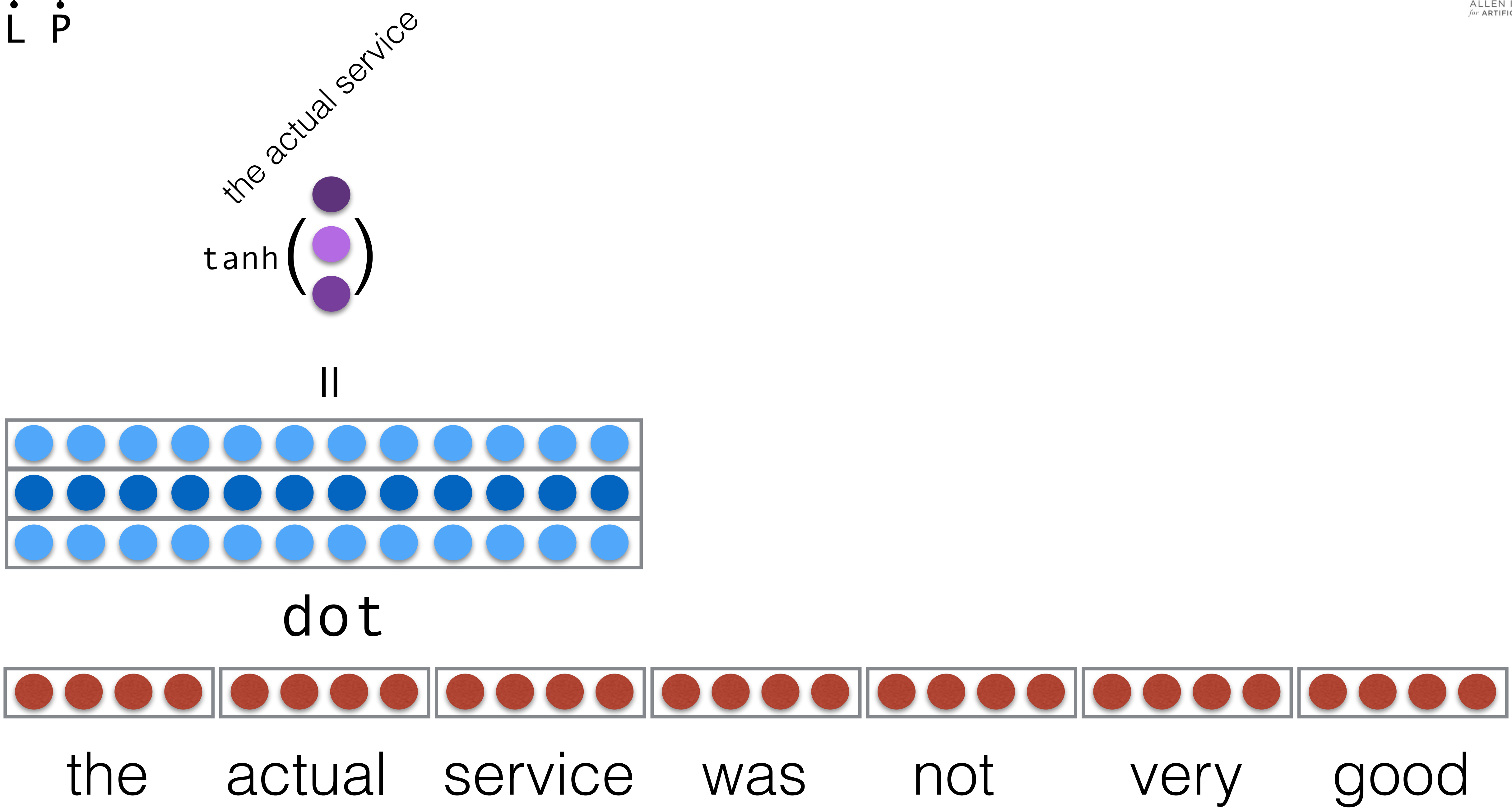
the actual
actual service
service was
was not
not very
very good
=
dot
the
actual
service
was
not
very
good



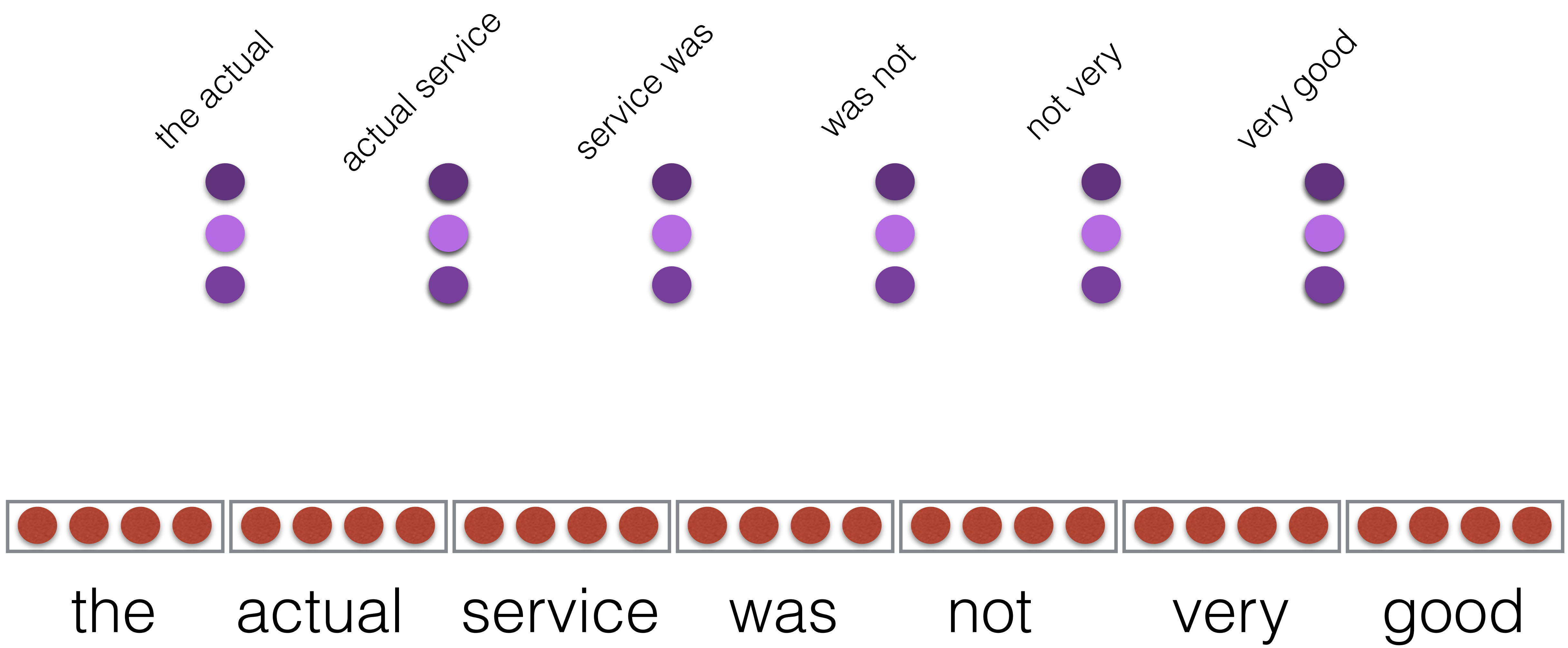
(usually also add non linearity)



(can have larger filters)



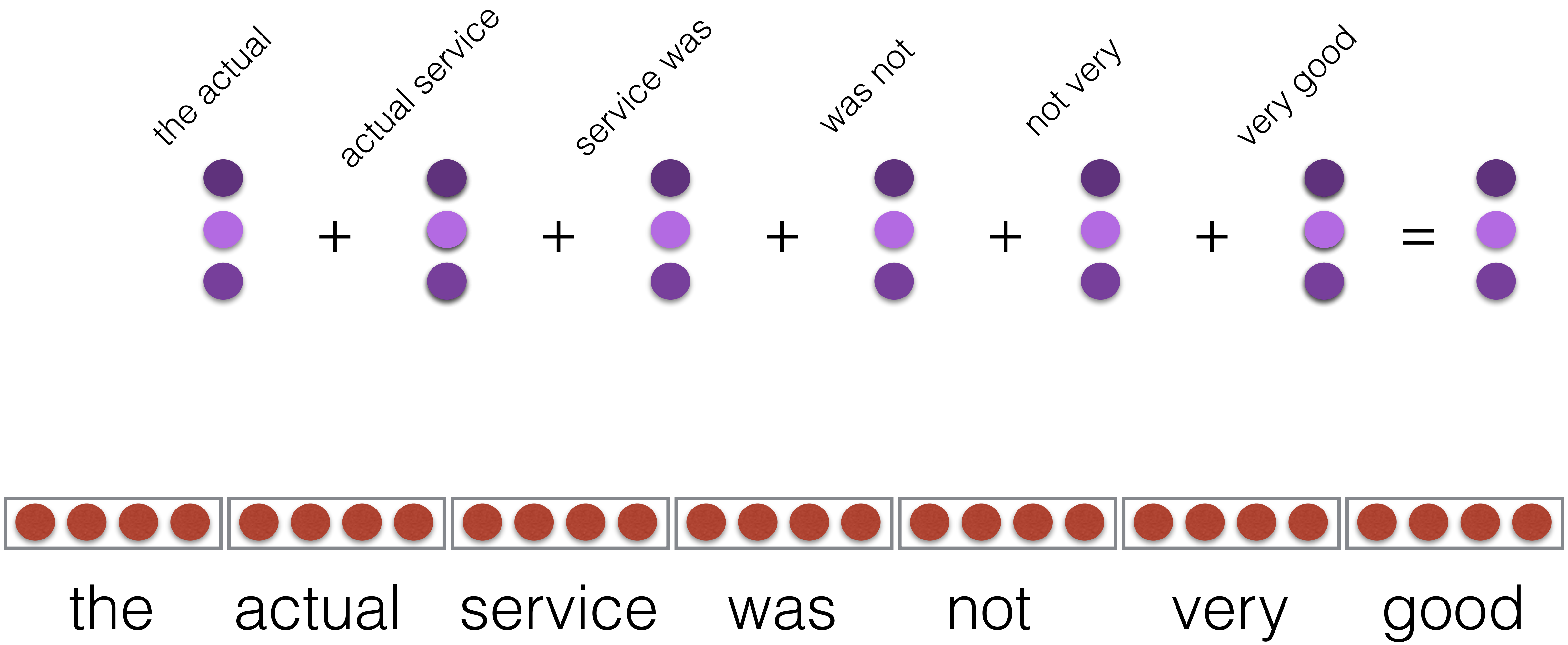
(can have larger filters)



we have the ngram vectors. now what?

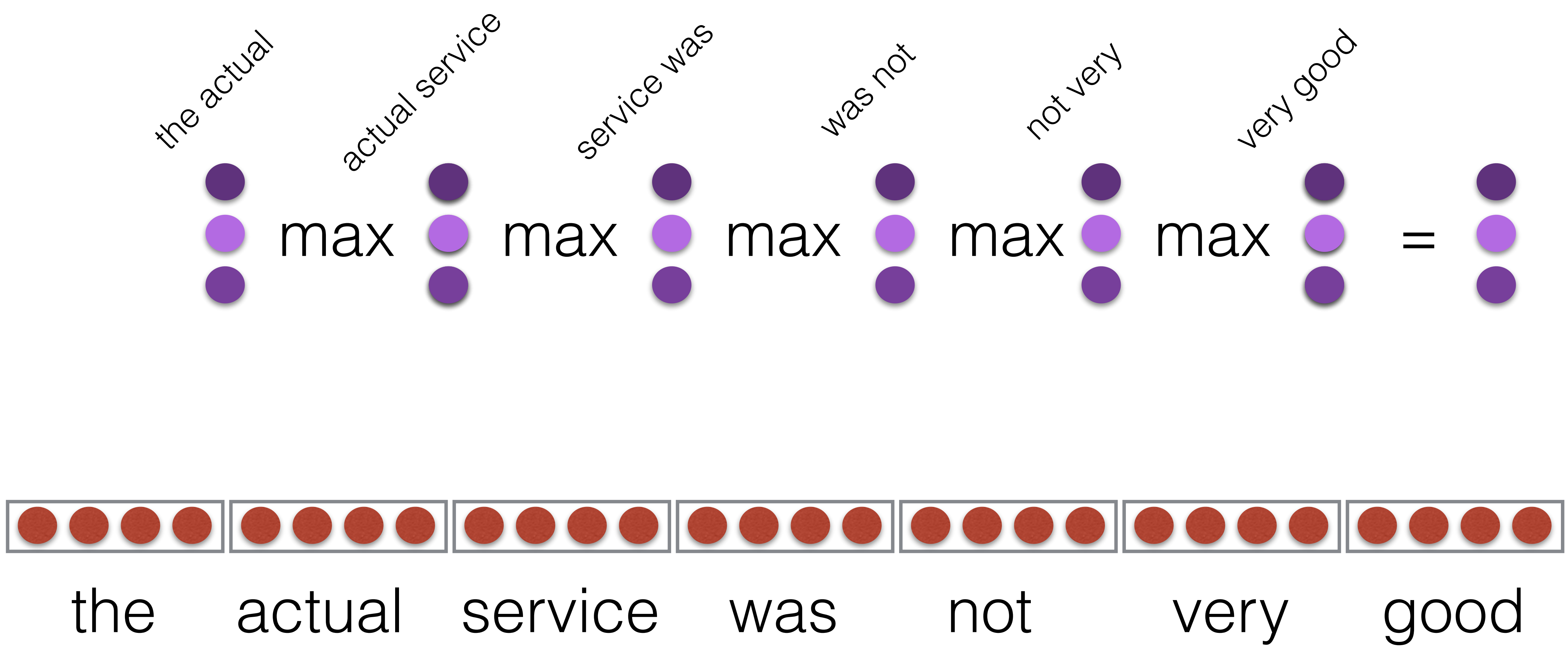
"Pooling"

Combine K vectors into a single vector



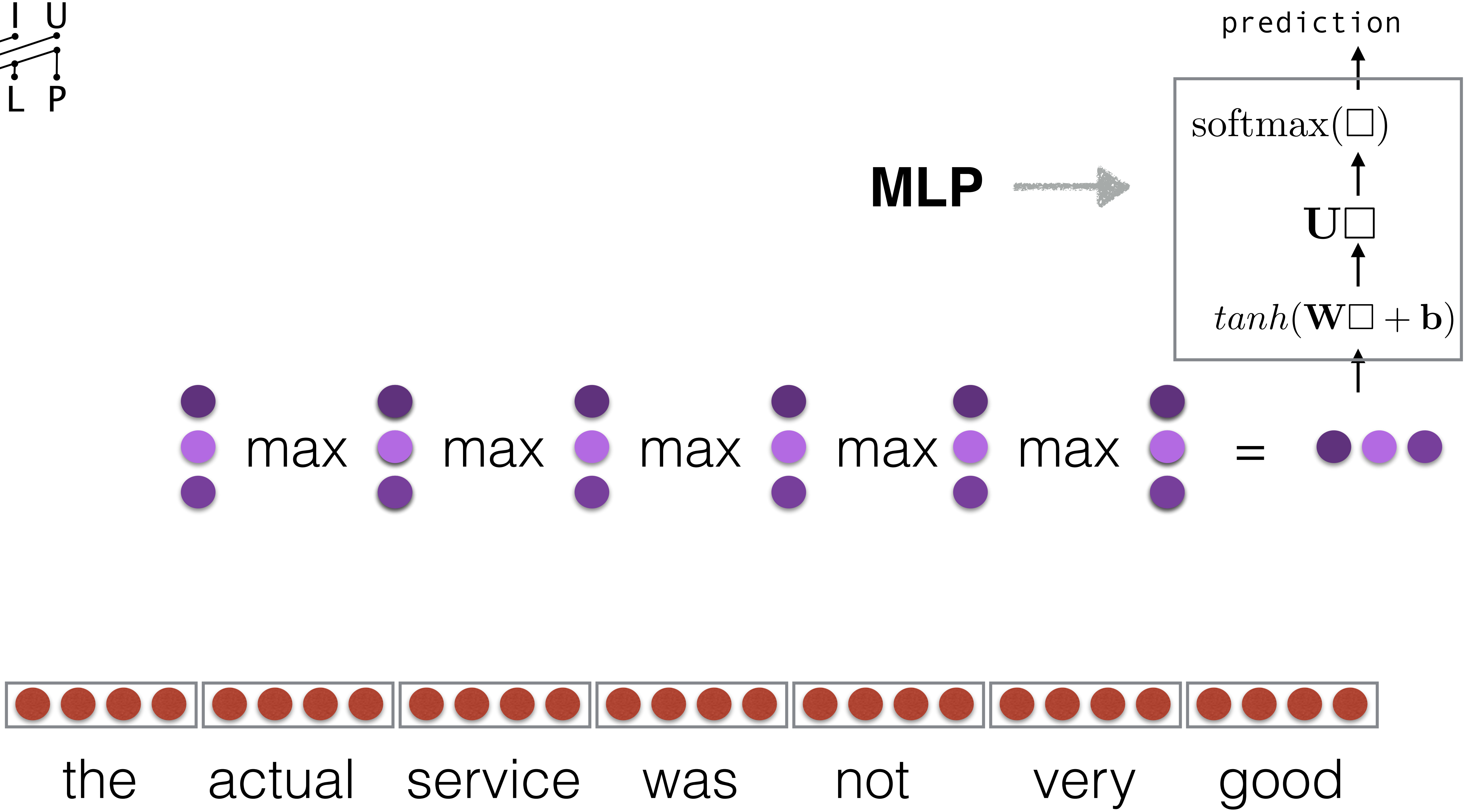
sum/avg pooling

average pooling



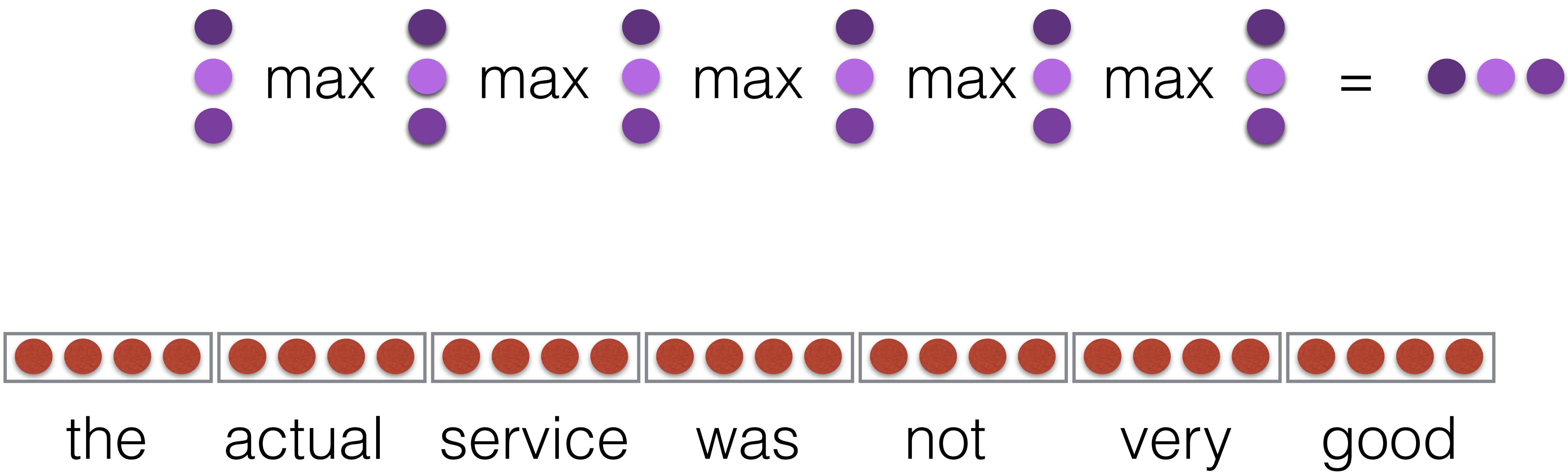
max pooling

(max in each coordinate)



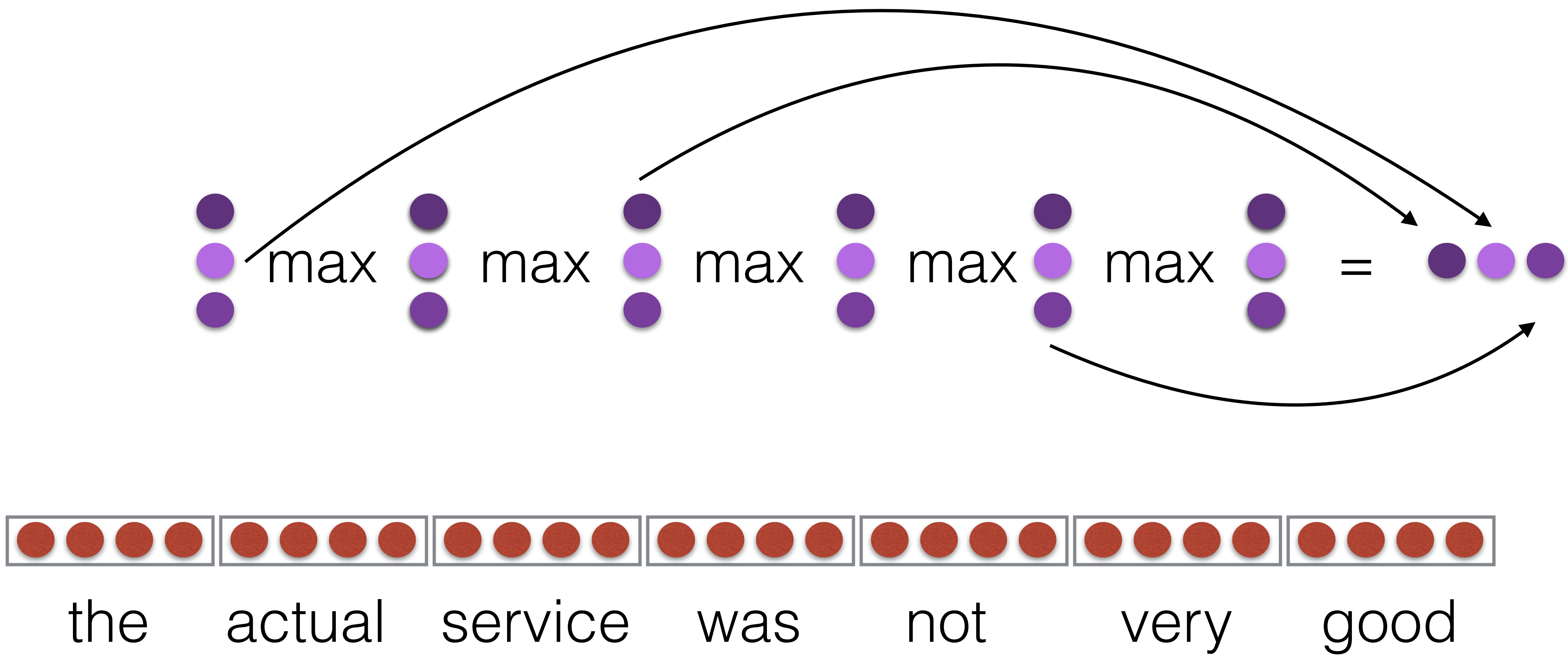
train end-to-end for some task

(train the MLP, the filter matrix, and the embeddings together)



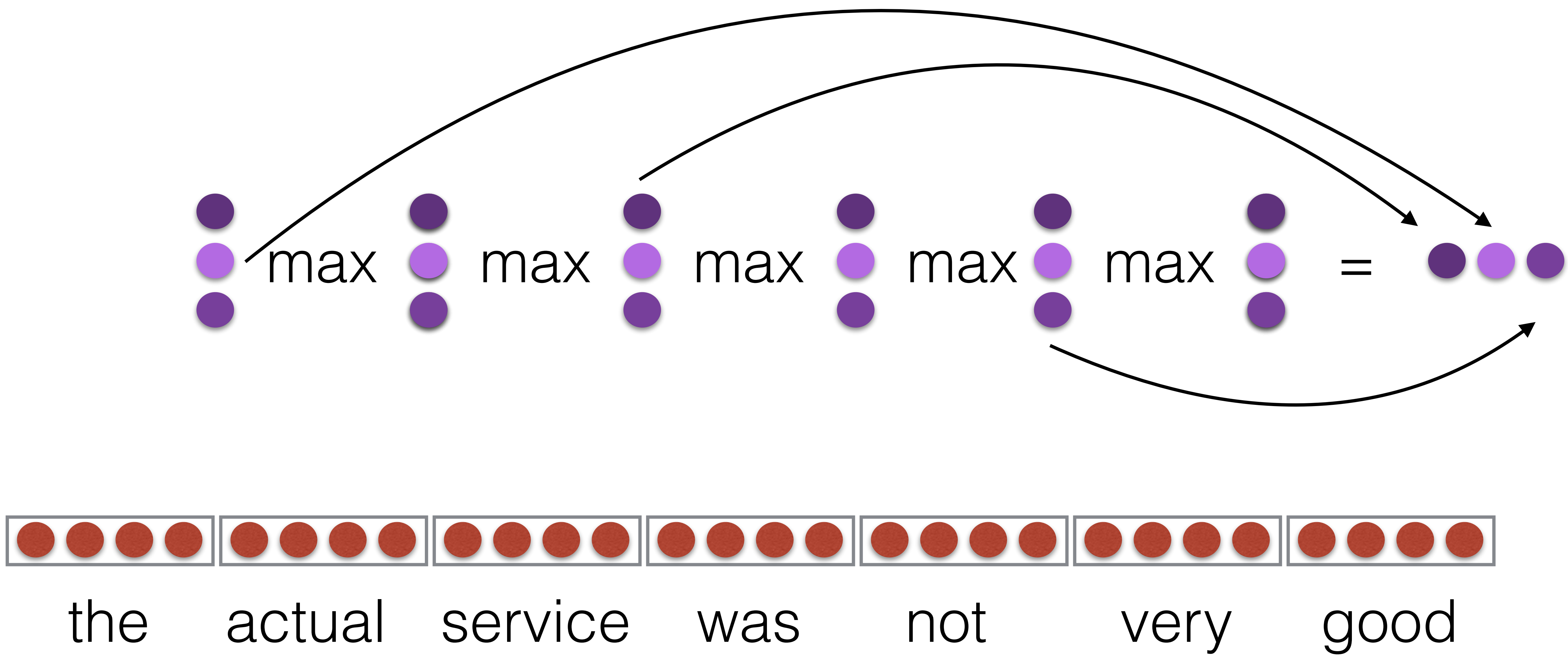
ngram detectors

each dimension comes from a specific ngram



ngram detectors

the filters act as "ngram detectors"
assigning high values to important ngrams



ngram detectors

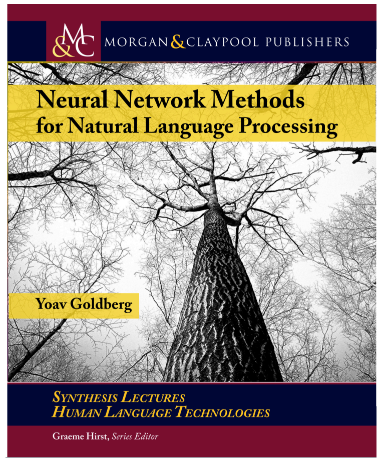
Textbook wisdom:

Each filter captures a group of **closely-related** ngrams

f_1
- *had no issues*
- *had zero issues*
- *had no problems*

f_2
- *is super cool*
- *was very interesting*
- *are well beyond*

- 300 filters → 300 families of ngrams
- Each filter is *homogeneous* - captures one family.



Textbook wisdom:

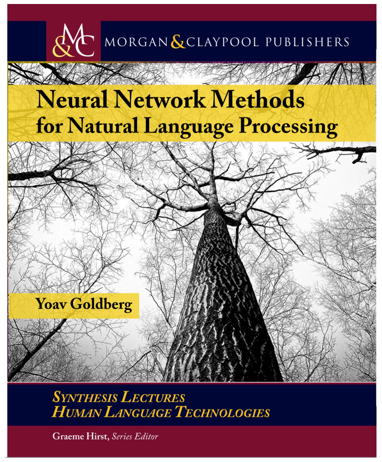
Each filter captures a group of **closely-related** ngrams

f_1
- *had no issues*
- *had zero issues*
- *had no problems*

f_2
- *is super cool*
- *was very interesting*
- *are well beyond*

- 300 filters → 300 families of ngrams
- Each filter is *homogeneous* - captures one family.

nope.

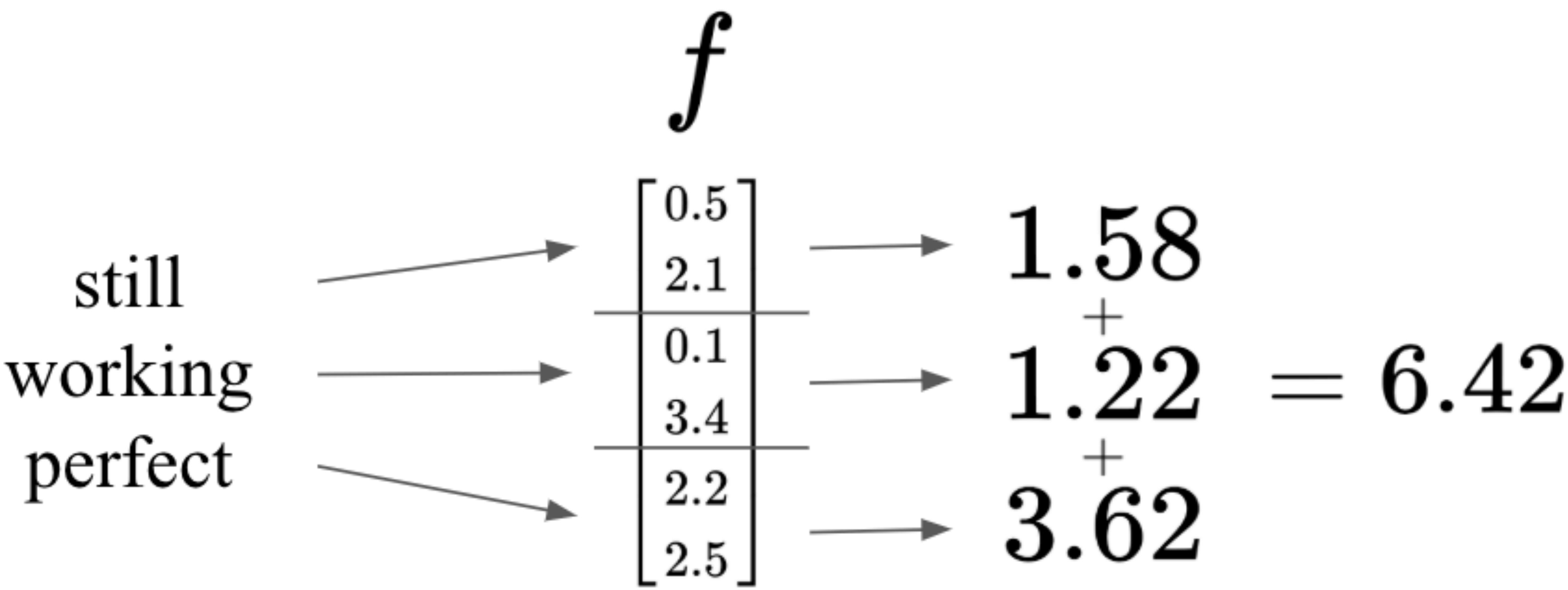


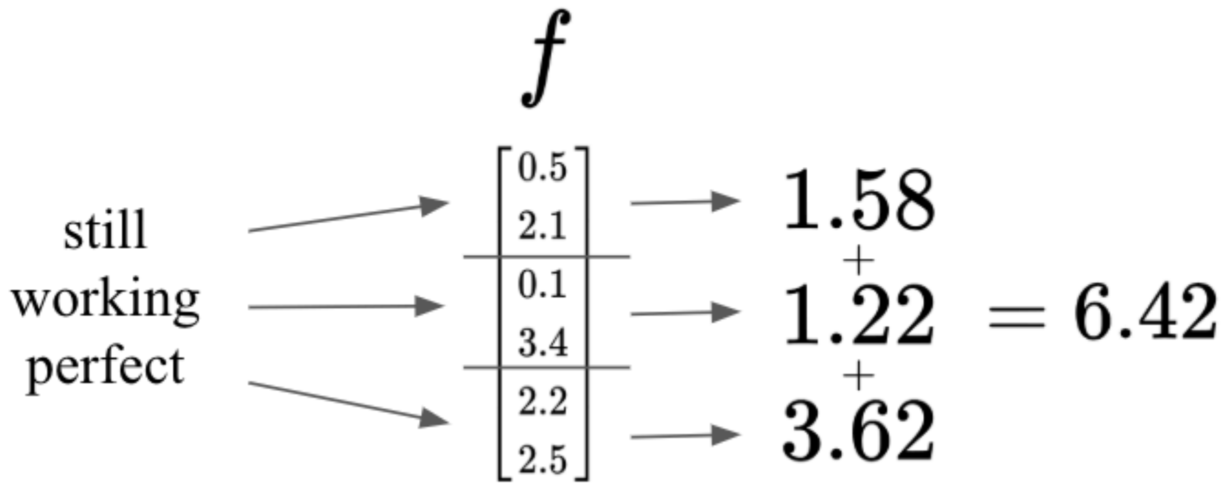
Understanding Convolutional Neural Networks for Text Classification

Alon Jacovi[1,2] **Oren Sar Shalom**[2,3] **Yoav Goldberg**[1,4]

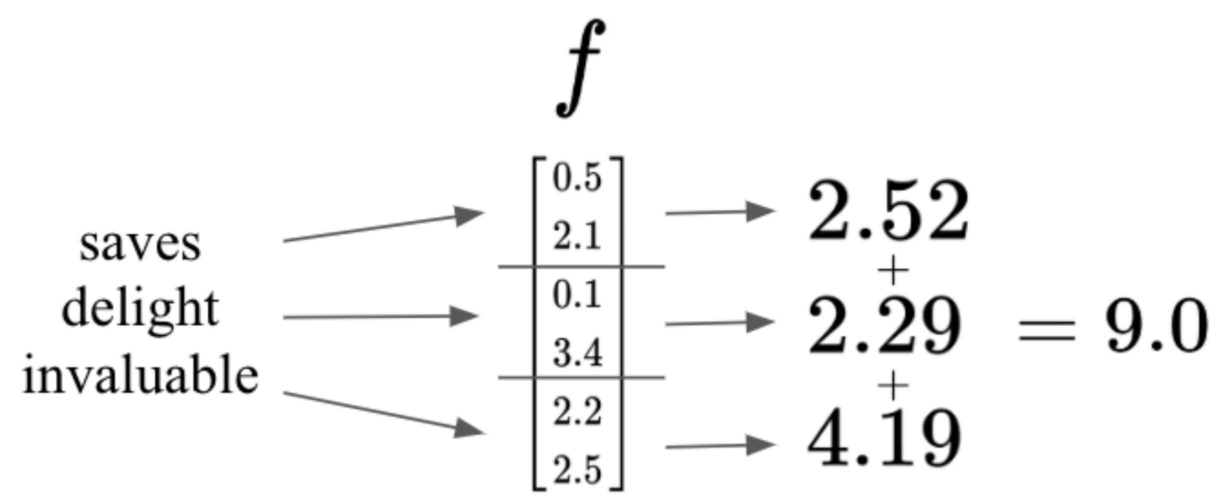
[1] Computer Science Department, Bar Ilan Univesity, Israel
[2] IBM Research, Haifa, Israel
[3] Intuit, Hod HaSharon, Israel
[4] Allen Institute for Artificial Intelligence
{alonjacovi,oren.sarshalom,yoav.goldberg}@gmail.com

$$\text{still working perfect} \longrightarrow \overset{f}{\begin{bmatrix}0.5\\2.1\\0.1\\3.4\\2.2\\2.5\end{bmatrix}} \longrightarrow 6.42$$


f
still
working
perfect
0.5
2.1
0.1
3.4
2.2
2.5
1.58
+
1.22
+
3.62
= 6.42



We can generate the ngrams that maximize each filter slot separately:



The generated maximized ngrams score **much higher** than the top ngrams.

filter	top ngram	score	top word for each slot	score
f1	poorly designed junk	7.31	poorly displaying landfill	**10.28**
f2	utterly useless .	6.33	stopped refund disabled	**7.96**
f3	still working perfect	6.42	saves delight invaluable	**9.0**
f4	a minor drawback	6.11	workstation high-quality drawback	**9.27**
f5	deserves four stars	5.56	excelente crossover incredible	**7.78**

max from corpus ngrams

max in each word

The generated maximized ngrams score **much higher** than the top ngrams.

filter	top ngram	score	top word for each slot	score
f1	poorly designed junk	7.31	poorly displaying landfill	**10.28**
f2	utterly useless .	6.33	stopped refund disabled	**7.96**
f3	still working perfect	6.42	saves delight invaluable	**9.0**
f4	a minor drawback	6.11	workstation high-quality drawback	**9.27**
f5	deserves four stars	5.56	excelente crossover incredible	**7.78**

max from corpus ngrams

max in each word

WHY????

The generated maximized ngrams score **much higher** than the top ngrams.

filter	top ngram	score	top word for each slot	score
f1	poorly designed junk	7.31	poorly displaying landfill	**10.28**
f2	utterly useless .	6.33	stopped refund disabled	**7.96**
f3	still working perfect	6.42	saves delight invaluable	**9.0**
f4	a minor drawback	6.11	workstation high-quality drawback	**9.27**
f5	deserves four stars	5.56	excelente crossover incredible	**7.78**

max from corpus ngrams

max in each word

	top ngrams				
rank	ngram	score	slot scores		
1	still working **perfect**	6.42	1.58	1.22	**3.62**
2	**works** - **perfect**	5.78	**1.91**	0.25	**3.62**
3	isolation proves **invaluable**	5.61	0.39	1.03	**4.19**
4	still near **perfect**	5.6	1.58	0.4	**3.62**
5	still working great	5.45	1.58	1.22	2.65
6	**works** as good	5.44	**1.91**	1.45	2.08
7	still **holding** strong	5.37	1.58	**1.81**	1.98

only *some* of the words maximize their slot scores

New concept: Slot Activation Pattern

Low Medium High

List of top-scoring ngrams for a specific filter

ngram	slot #1	slot #2	slot #3
was super intriguing	1.01	3.16	5.84
go wrong pairing	3.97	4.12	1.65
am so grateful	2.59	3.27	4.07
overall very worth	3.84	1.86	4.22
go wrong bringing	3.97	4.12	1.81
also well worth	1.83	3.06	4.22
- super compassionate	0.51	3.17	5.01
go wrong when	3.97	4.12	-0.4
a well oiled	0.75	3.06	4.84

New concept: Slot Activation Pattern

Low Medium High

List of top-scoring ngrams
for a specific filter

ngram	slot #1	slot #2	slot #3
was super intriguing	1.01	3.16	5.84
go wrong pairing	3.97	4.12	1.65
am so grateful	2.59	3.27	4.07
overall very worth	3.84	1.86	4.22
go wrong bringing	3.97	4.12	1.81
also well worth	1.83	3.06	4.22
- super compassionate	0.51	3.17	5.01
go wrong when	3.97	4.12	-0.4
a well oiled	0.75	3.06	4.84

New concept: Slot Activation Pattern

List of top-scoring ngrams
for a specific filter

ngram	slot #1	slot #2	slot #3
was super intriguing	1.01	3.16	5.84
go wrong pairing	3.97	4.12	1.65
am so grateful	2.59	3.27	4.07
overall very worth	3.84	1.86	4.22
go wrong bringing	3.97	4.12	1.81
also well worth	1.83	3.06	4.22
- super compassionate	0.51	3.17	5.01
go wrong when	3.97	4.12	-0.4
a well oiled	0.75	3.06	4.84

New concept: Slot Activation Pattern

ngram	slot #1	slot #2	slot #3
was super intriguing	1.01	3.16	5.84
go wrong pairing	3.97	4.12	1.65
am so grateful	2.59	3.27	4.07
overall very worth	3.84	1.86	4.22
go wrong bringing	3.97	4.12	1.81
also well worth	1.83	3.06	4.22
- super compassionate	0.51	3.17	5.01
go wrong when	3.97	4.12	-0.4
a well oiled	0.75	3.06	4.84

Cluster filter ngrams according to slot activations

Each cluster is a homogeneous family of ngrams.

The same filter detected both families.

cluster 1

cluster 2

ngram	slot #1	slot #2	slot #3
centroid	0.75	1.97	2.79
was super intriguing	1.01	3.16	5.84
am so grateful	2.59	3.27	4.07
overall very worth	3.84	1.86	4.22
also well worth	1.83	3.06	4.22
- super compassionate	0.51	3.17	5.01
a well oiled	0.75	3.06	4.84
centroid	2.87	2.17	0.12
go wrong bringing	3.97	4.12	1.81
go wrong pairing	3.97	4.12	1.65
go wrong when	3.97	4.12	-0.4

Cluster filter ngrams according to slot activations

Each cluster is a homogeneous family of ngrams.

The same filter detected both families.

cluster 1

cluster 2

	ngram	slot #1	slot #2	slot #3
	centroid	0.75	1.97	2.79
cluster 1	was super intriguing	1.01	3.16	5.84
cluster 1	am so grateful	2.59	3.27	4.07
cluster 1	overall very worth	3.84	1.86	4.22
cluster 1	also well worth	1.83	3.06	4.22
cluster 1	- super compassionate	0.51	3.17	5.01
cluster 1	a well oiled	0.75	3.06	4.84
	centroid	2.87	2.17	0.12
cluster 2	go wrong bringing	3.97	4.12	1.81
cluster 2	go wrong pairing	3.97	4.12	1.65
cluster 2	go wrong when	3.97	4.12	-0.4

filters are **not homogenous**

a filter may detect **multiple families** of ngrams

complex behavior.

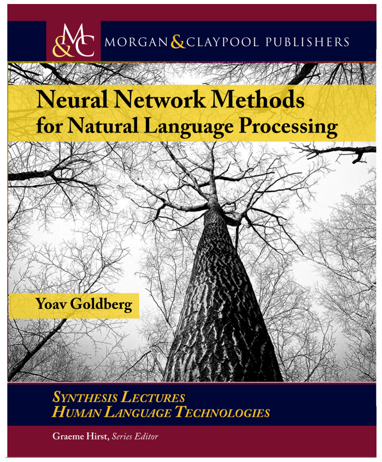
300 filters --> more than 300 ngram types.

filters are **not homogenous**

a filter may detect **multiple families** of ngrams

Textbook wisdom 2:

filters detect the **presence**
of specific ngrams / words



2.59 'm 1.86 really 5.05 pleased

High-scoring ngram

does slot #2 capture the word "*really*"?

weak score

2.59 ~1.86 5.05

'm _____ pleased

1.86 is an average score for slot #2.
many words get similar scores

weak score

2.59 ~1.86 5.05

'm _____ pleased

1.86 is an average score for slot #2.
many words get similar scores

slot #2 is a wildcard slot?

weak score

2.59	~1.86	5.05
'm	_____	pleased

1.86 is an average score for slot #2.
many words get similar scores

slot #2 is a wildcard slot?

nope.

Strong **negative** score

-3.4

'm not pleased

slot #2 is a wildcard slot?

nope.

Strong **negative** score

-3.4

'm not pleased

slot #2 is detecting
the **absence** of the word not

1D ConvNets are Complex

Intuition

Each filter detects
a family of ngrams

Filters detect presence

1D ConvNets are Complex

Intuition	**Real World**
Each filter detects a family of ngrams	Some filters detect multiple families of ngrams
Filters detect presence	Some filters detect absence

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

also look at:

Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context

Urvashi Khandelwal, He He, Peng Qi, Dan Jurafsky
Computer Science Department
Stanford University
{urvashik,hehe,pengqi,jurafsky}@stanford.edu

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what can't they do?

Q5: What is the representation power of diff archs?

Q6: Extracting a discrete reps from a trained model.

Q2: What is encoded/captured in a vector?

Q2: What is encoded/captured in a vector?

Published as a conference paper at ICLR 2017

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Yossi Adi[1,2], Einat Kermany[2], Yonatan Belinkov[3], Ofer Lavi[2], Yoav Goldberg[1]

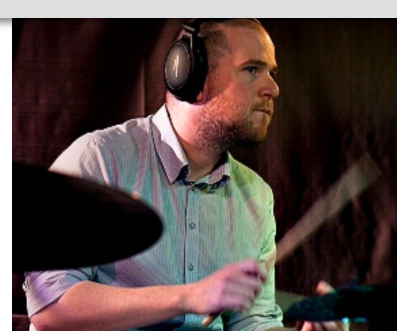


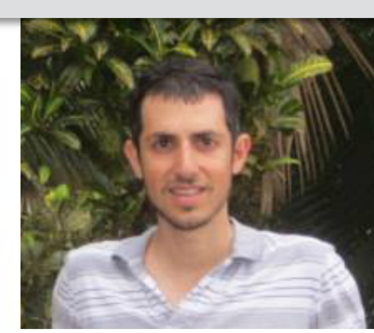
Q2: What is encoded/captured in a vector?

Published as a conference paper at ICLR 2017

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Methodology: can you train a classifier to predict X from the representation?





What's in a sentence?

To fully reconstruct a sentence, we need to know:

- How many words?
- Which words?
- What order?

Compare different sentence representations based on their preservation of these properties.

Formulate as Prediction Tasks

Sentence Length

Word order

Which words?

Formulate as Prediction Tasks

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (8 bins)

Word order

Which words?

Formulate as Prediction Tasks

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (8 bins)

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **a**?

Word order

Formulate as Prediction Tasks

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (8 bins)

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **a**?

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)

dim	acc
100	
300	
500	
750	
1000	

Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)

dim	acc
100	50%
300	80%
500	82%
750	79%
1000	83%

Some Results

Sentence Length

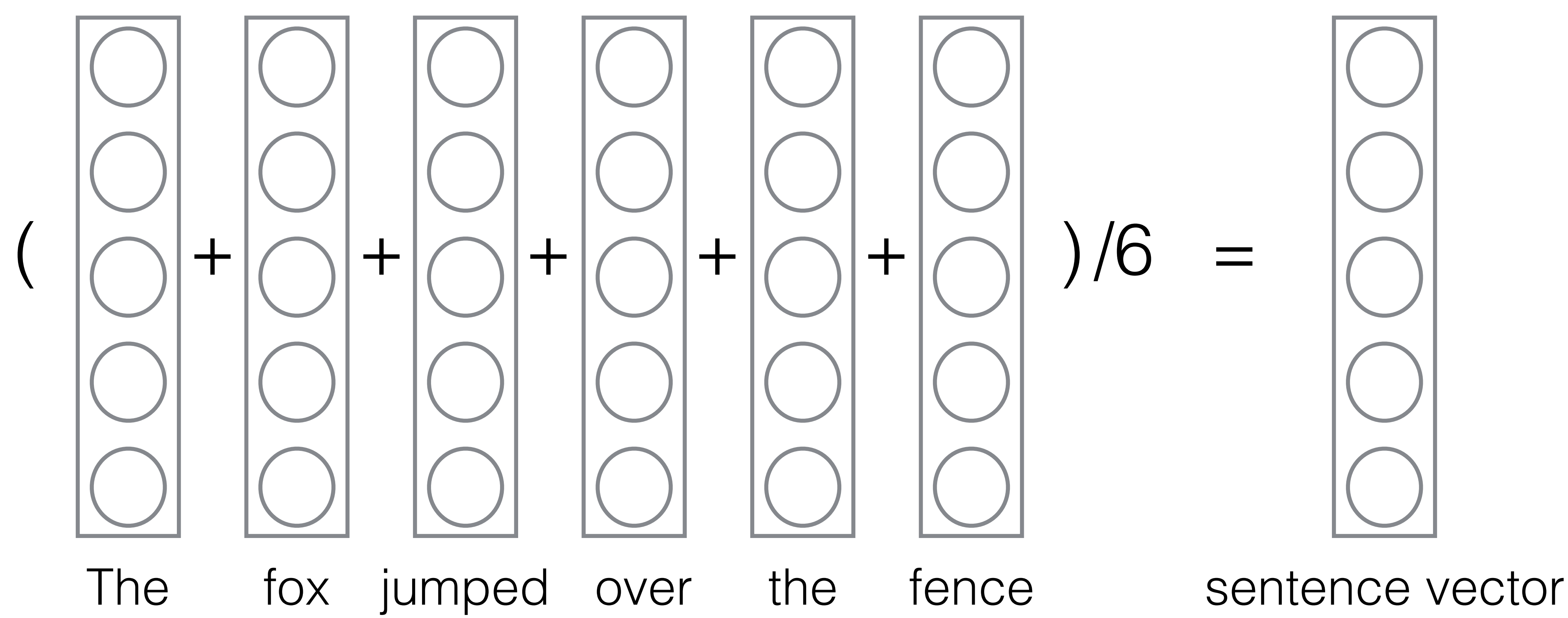
Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)		CBOW
dim	acc	
100	50%	??
300	80%	
500	82%	
750	79%	
1000	83%	

CBOW (Continuous-Bag-of-Words)

- Represent each word in the sentence as a vector (word2vec)
- The average of these vectors is the sentence vector



Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)		CBOW
dim	acc	
100	50%	??
300	80%	
500	82%	
750	79%	
1000	83%	

Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)		CBOW
dim	acc	
100	50%	45%
300	80%	49%
500	82%	57%
750	79%	60%
1000	83%	60%

Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM)		CBOW
dim	acc	
100	50%	45%
300	80%	49%
500	82%	57%
750	79%	60%
1000	83%	60%

surprisingly high accuracy for 8-class classification,
considering that CBOW is an averaged representation

Some Results

Sentence Length

Input:
Sentence encoding.
Task:
Predict length (binned)

Baseline 22%

Encoder (LSTM) dim	Encoder (LSTM) acc	CBOW
100	50%	45%
300	80%	49%
500	82%	57%
750	79%	60%
1000	83%	60%

CBOW encodes length??

surprisingly high accuracy for 8-class classification,
considering that CBOW is an averaged representation

BIU NLP

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

Some Results

reviewer 2:

```
The paper reads very well, but
a) I do not understand the motivation, and
b) the experiments seem flawed.
```

The average over CBOW word embeddings should never encode for sentence length. The fact that you learn reasonably well with these representations, suggest overfitting. This may well be, since Wikipedia contains tons of duplicate or near-duplicate sentences.

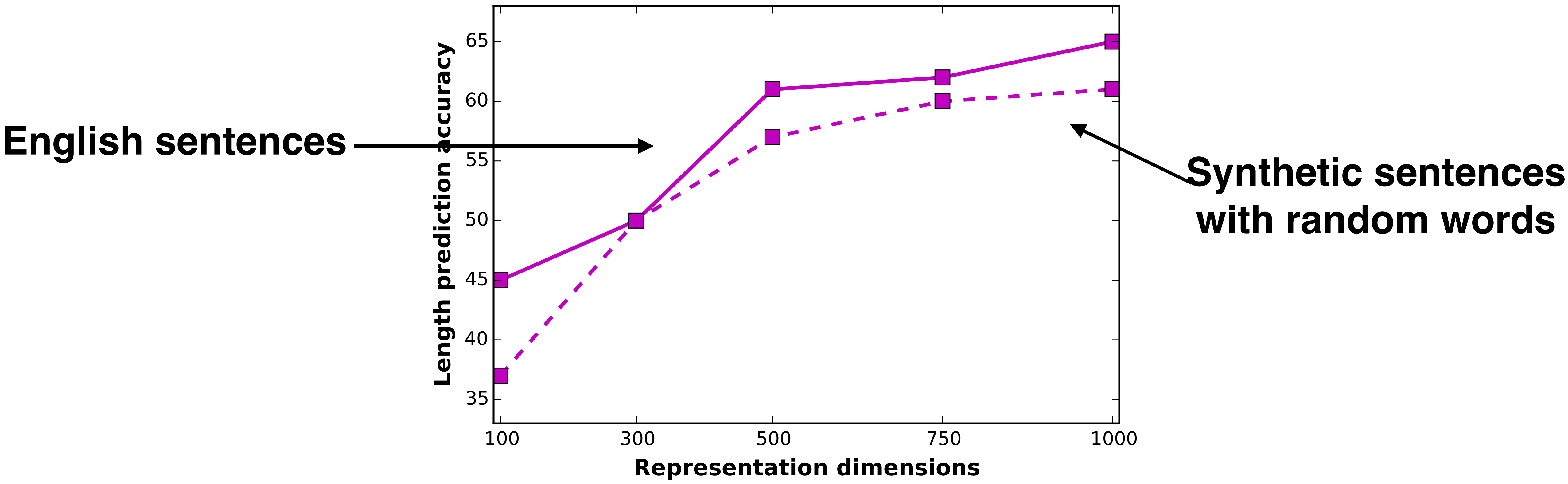
considering that CBOW is an averaged representation

How does CBOW encode length?

- Maybe some words are predictive of longer sentences?

How does CBOW encode length?

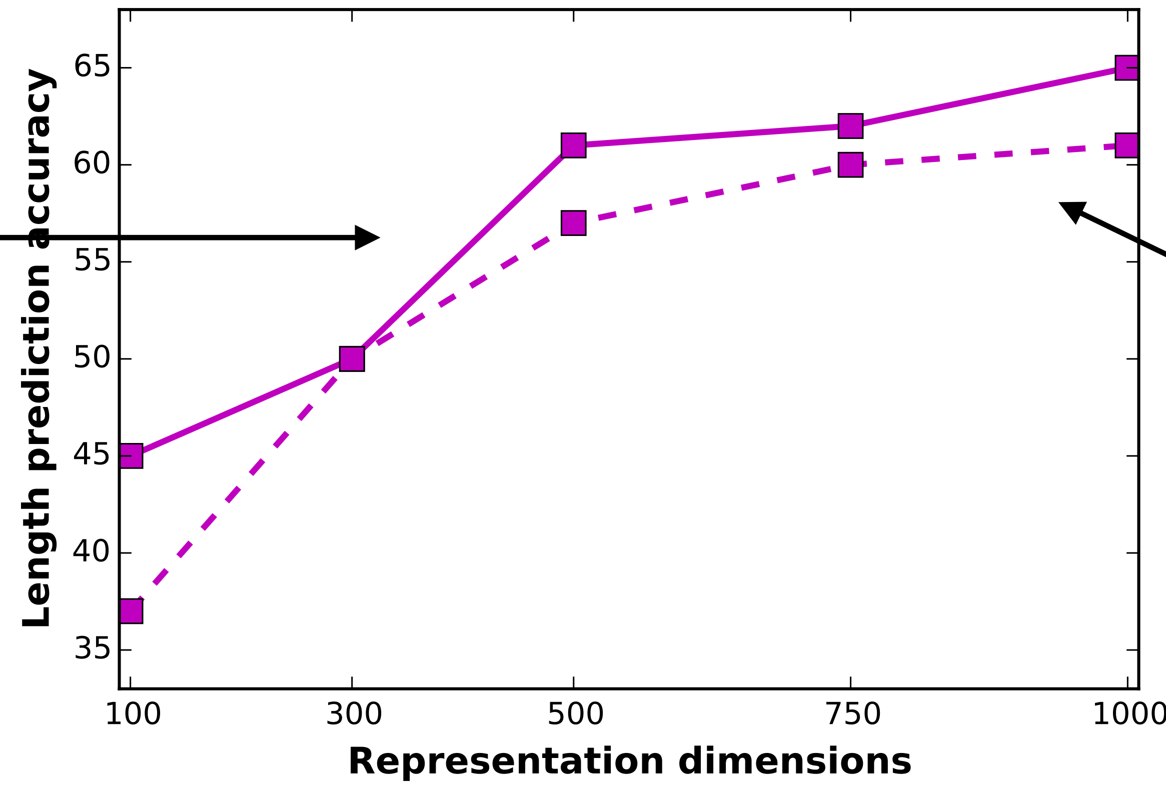
- Maybe some words are predictive of longer sentences?



How does CBOW encode length?

- Maybe some words are predictive of longer sentences?

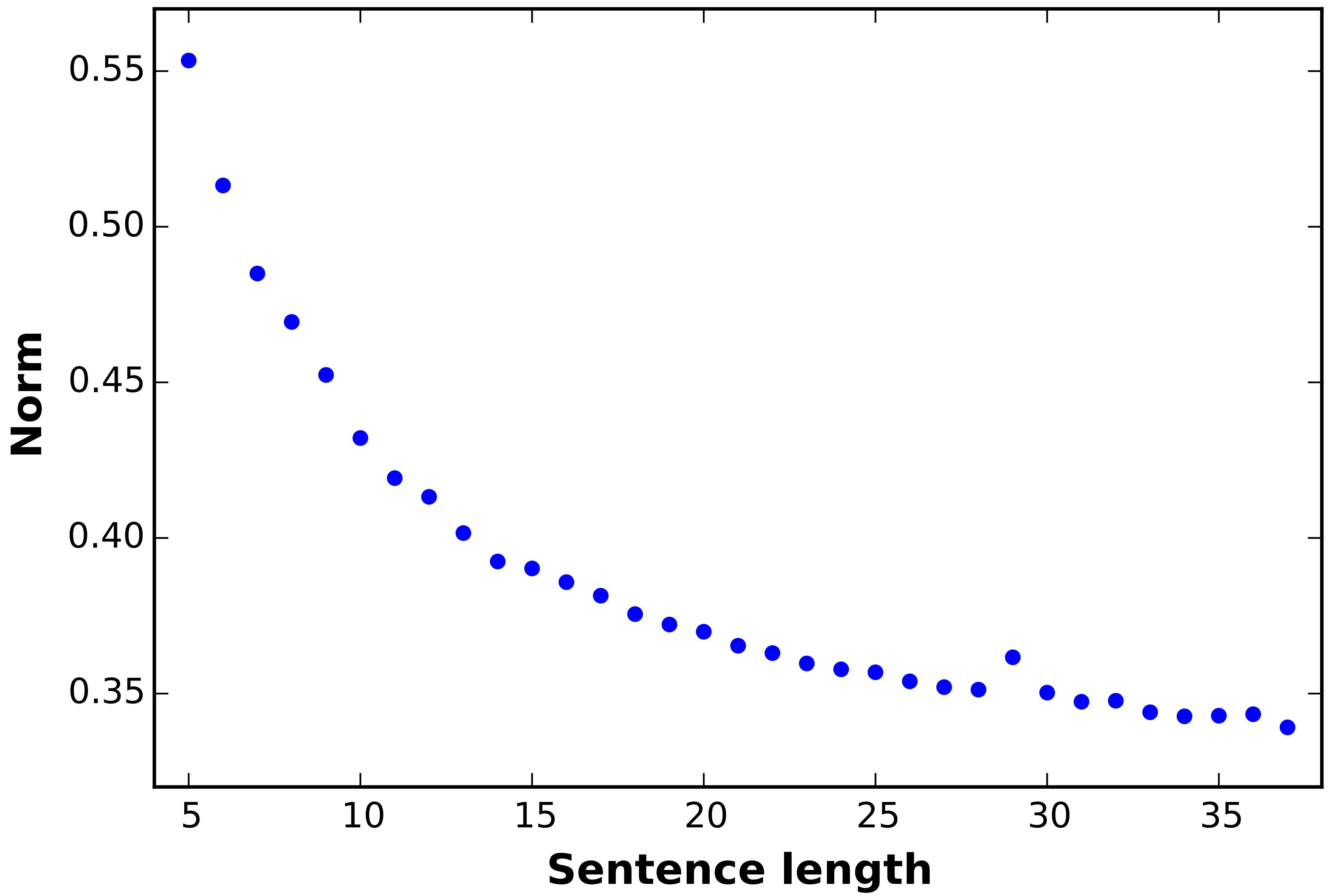
English sentences



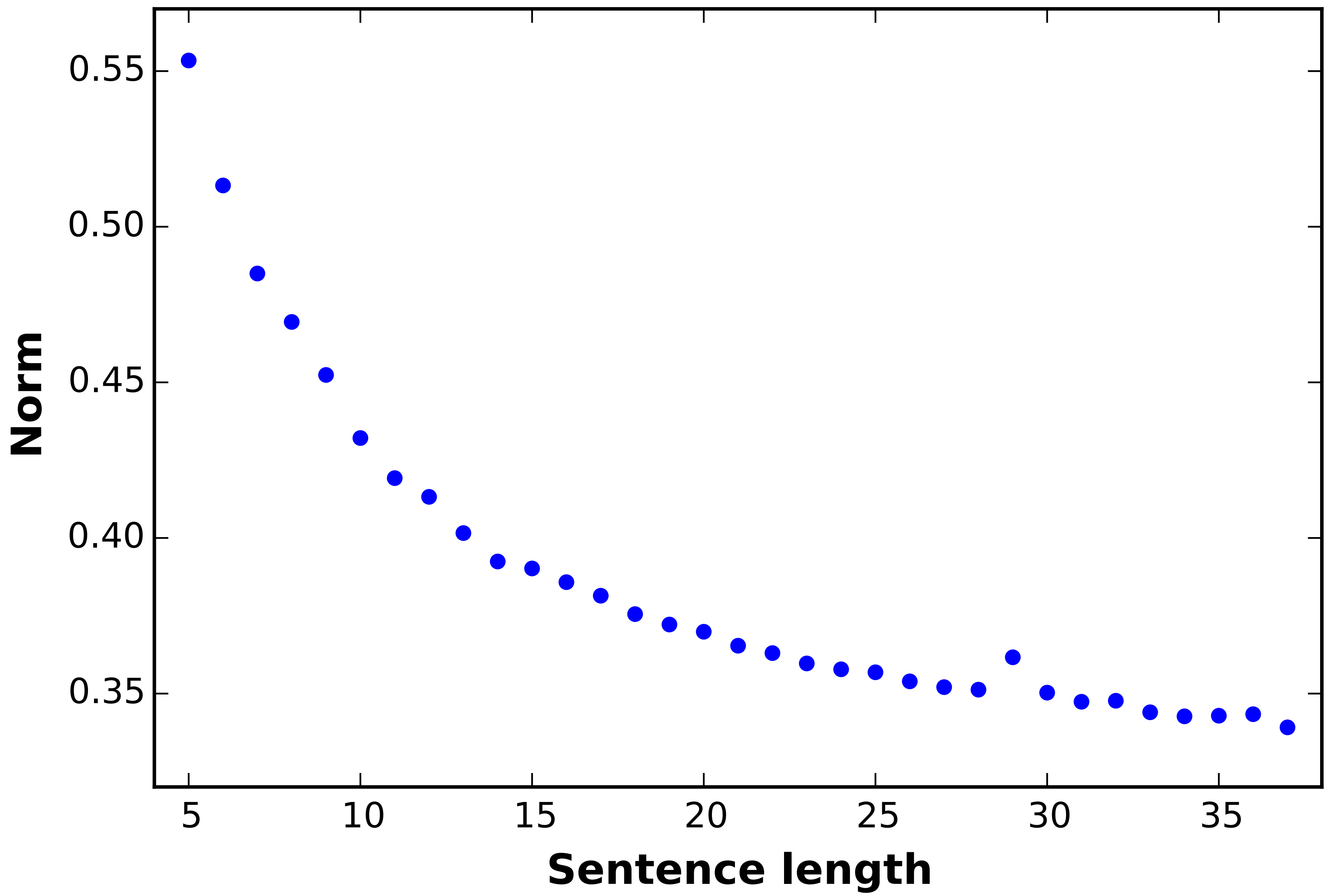
Synthetic sentences with random words

We do have an explanation!

How does CBOW encode length?



How does CBOW encode length?



(Why?)

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM)		CBOW
dim	acc	
100		
300		
500		
750		
1000		

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM)		CBOW
dim	acc	
100	70%	
300	75%	
500	76%	
750	**80%**	
1000	75%	

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM)		CBOW
dim	acc	
100	70%	
300	75%	
500	76%	
750	**80%**	
1000	75%	

higher dim not necessarily better!
(reconstruction BLEU does improve in higher dims)

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM) CBOW

dim	acc
100	70%
300	75%
500	76%
750	**80%**
1000	75%

power moves to the decoder (which we throw away)

reconstruction BLEU does improve in higher dims

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM)		CBOW
dim	acc	
100	70%	**84%**
300	75%	**88%**
500	76%	60%
750	**80%**	60%
1000	75%	60%

Some Results

Which words?

Input:
Sentence encoding **s**.
Word encoding **a**.
Task:
Does **s** contain **w**?

Encoder (LSTM)		CBOW
dim	acc	
100	70%	**84%**
300	75%	**88%**
500	76%	60%
750	**80%**	60%
1000	75%	60%

cbow better at preserving sentence words

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

Encoder (LSTM)		CBOW
dim	acc	
100	79%	
300	83%	
500	85%	
750	86%	
1000	**90**%	

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

	Encoder (LSTM)	CBOW
dim	acc	
100	79%	70%
300	83%	70%
500	85%	66%
750	86%	66%
1000	**90**%	66%

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

	Encoder (LSTM)		CBOW
	dim	acc	wait what?
	100	79%	70%
	300	83%	70%
	500	85%	66%
	750	86%	66%
	1000	**90%**	66%

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

Encoder (LSTM) dim	Encoder (LSTM) acc	CBOW
		wait what?
100	79%	70%
300	83%	70%
500	85%	66%
750	86%	66%
1000	**90%**	66%

what if we trained on words alone, without sentence representation?

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

	Encoder (LSTM)		CBOW	
dim	acc		wait what?	
100	79%	67%	70%	67%
300	83%	67%	70%	68%
500	85%	67%	66%	65%
750	86%	67%	66%	64%
1000	**90**%	65%	66%	64%

what if we trained on words alone,
without sentence representation?

Some Results

Word order

Input:
Sentence encoding **s**.
Word encoding **a**.
Word encoding **b**.
Task:
Does **a** appear in **s** before **b**?

	Encoder (LSTM)		CBOW	
dim	acc		wait what?	
100	79%	67%	70%	67%
300	83%	67%	70%	68%
500	85%	67%	66%	65%
750	86%	67%	66%	64%
1000	**90**%	65%	66%	64%

word identities alone get you quite far,
but cbow still informative re order!

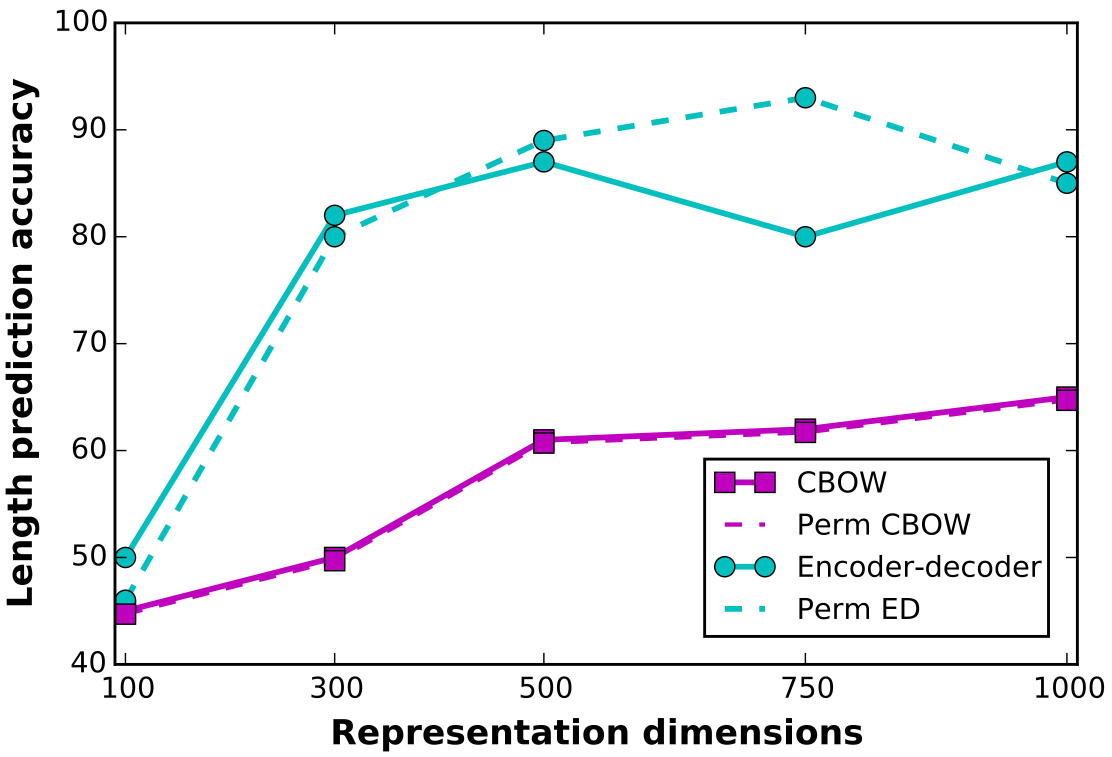
Does it Learn to Represent English or Just Sequences?

- We use the trained encoders
- But evaluate them on permuted sentences

encode(“fence over jumped the fox The”)

Does **fence** appear before **fox**?

Does it Learn to Represent English or Just Sequences?

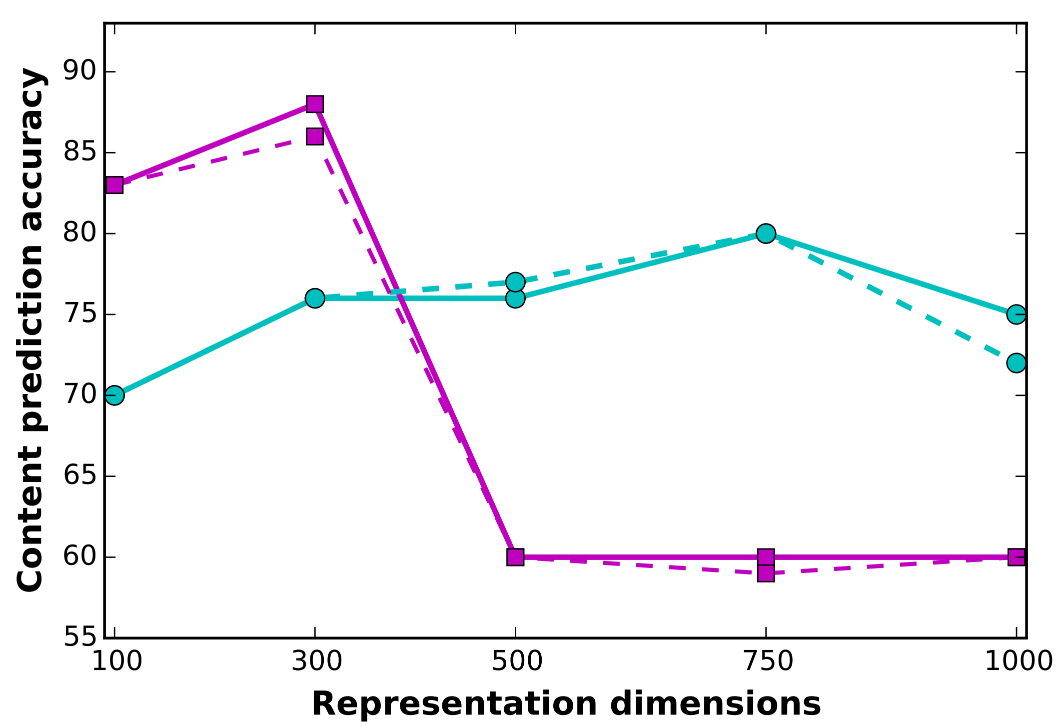


LSTM

cbow

Length Prediction

Does it Learn to Represent English or Just Sequences?

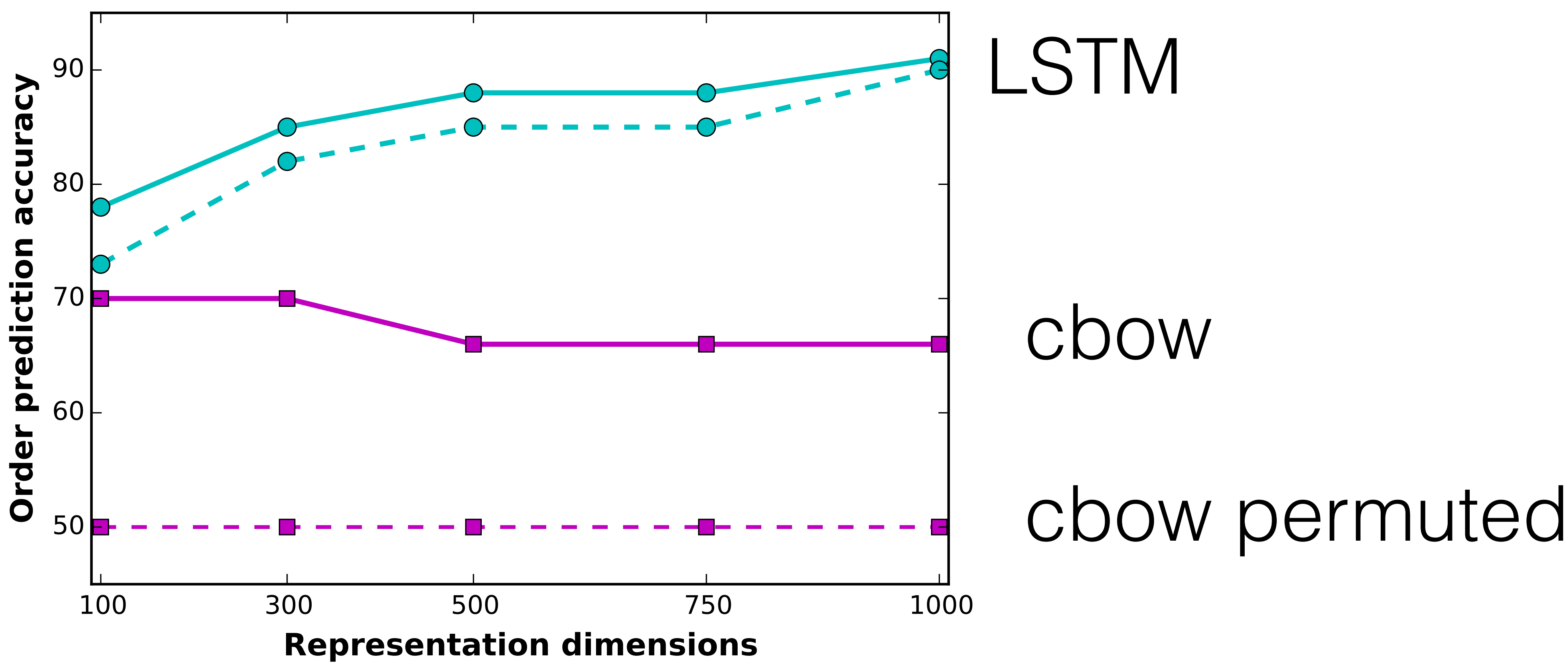


LSTM

cbow

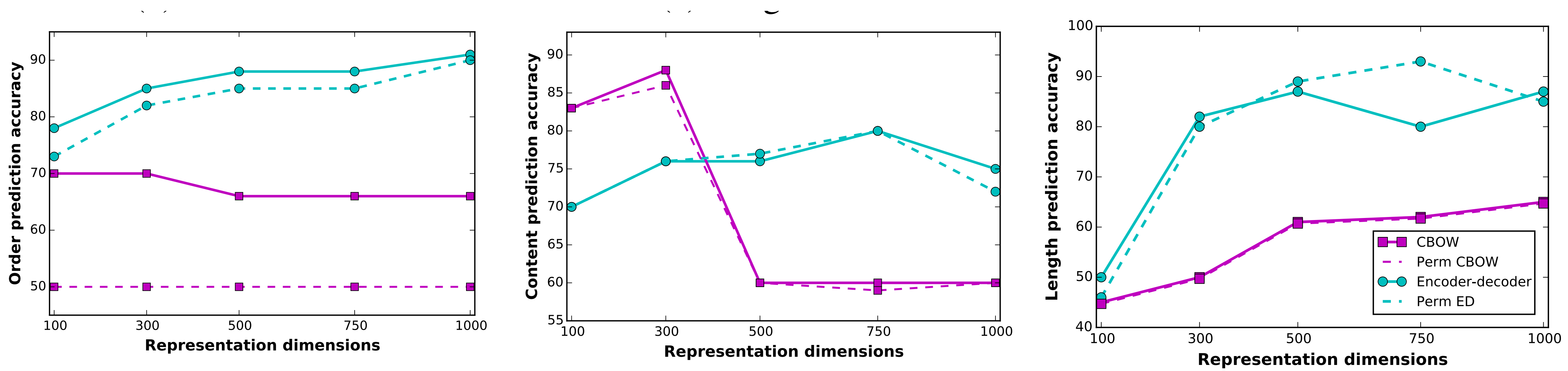
Content Prediction

Does it Learn to Represent English or Just Sequences?



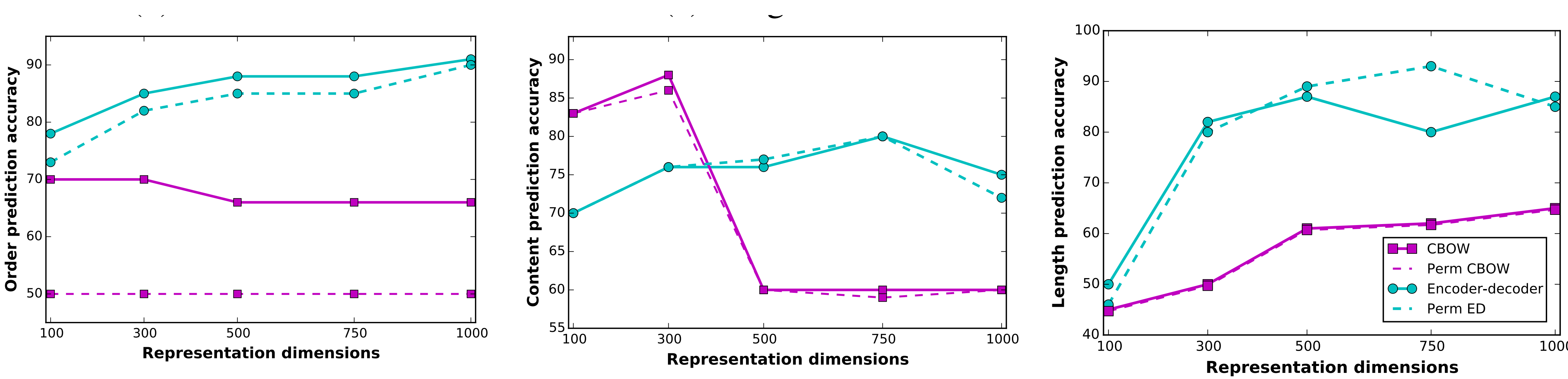
Order Prediction

Does it Learn to Represent English or Just Sequences?



auto-encoder LSTM
does not really care what it encodes.
a generic sequence encoder.

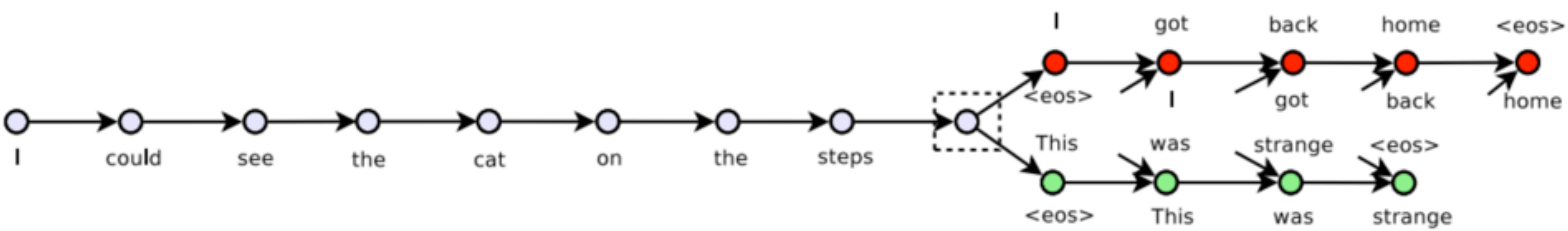
Does it Learn to Represent English or Just Sequences?



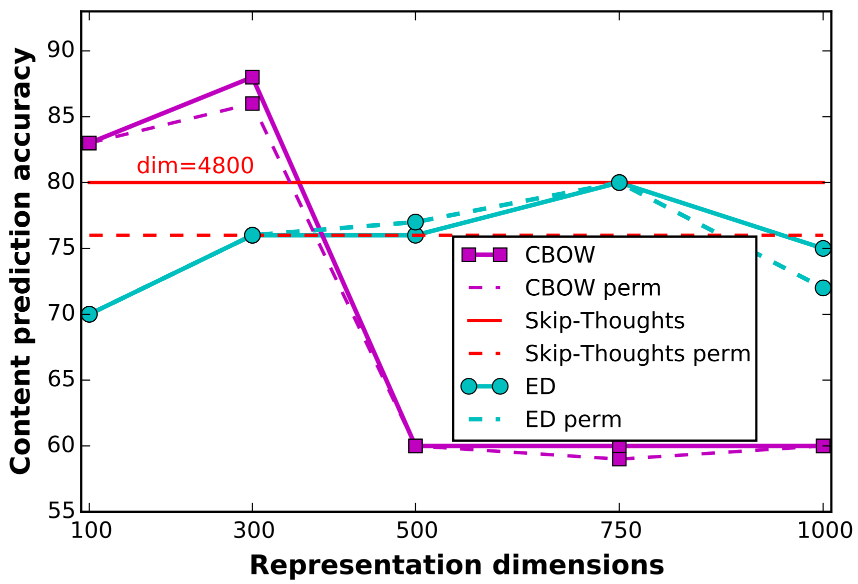
auto-encoder LSTM
does not really care what it encodes.
a generic sequence encoder.

nat-lang information is in the decoder.

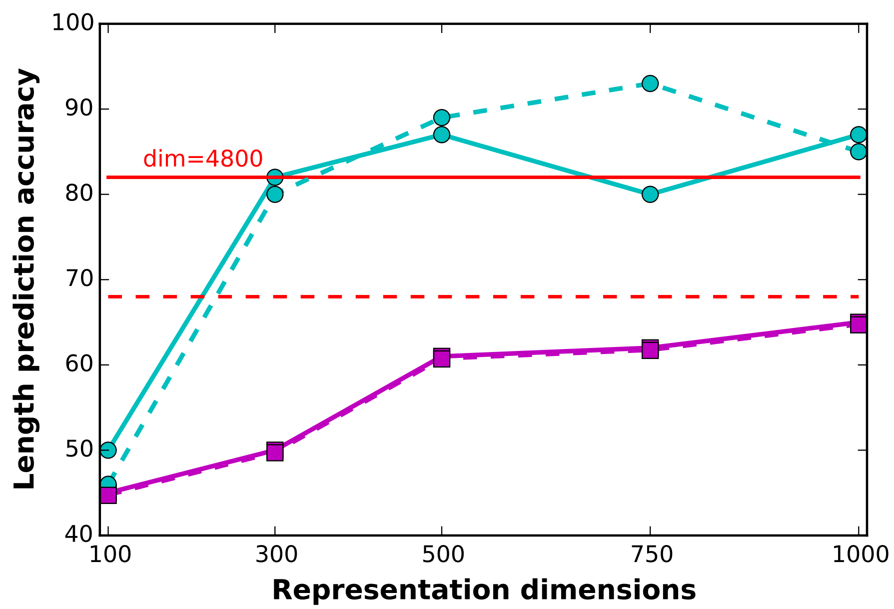
Skip-Thought Vectors



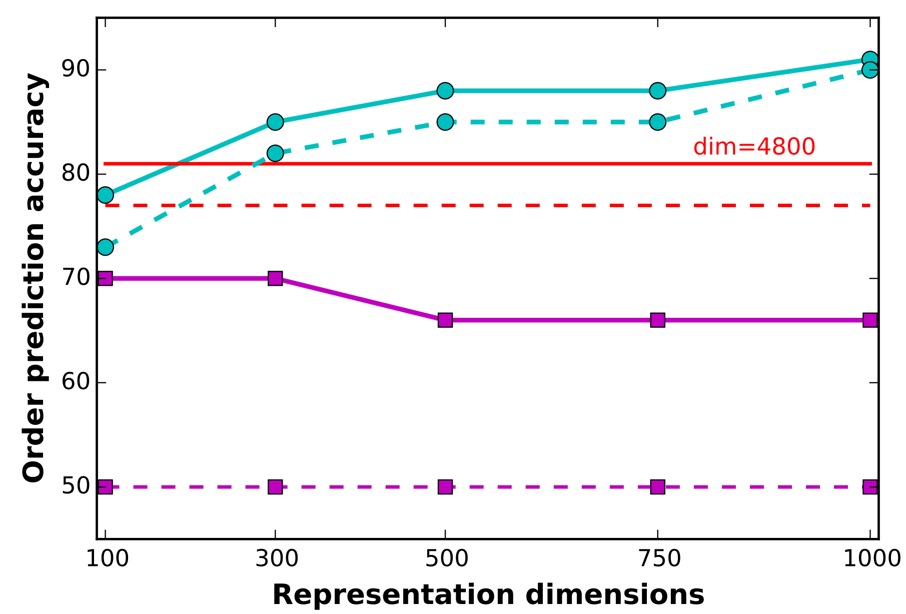
Does it Learn to Represent English or Just Sequences?



Content

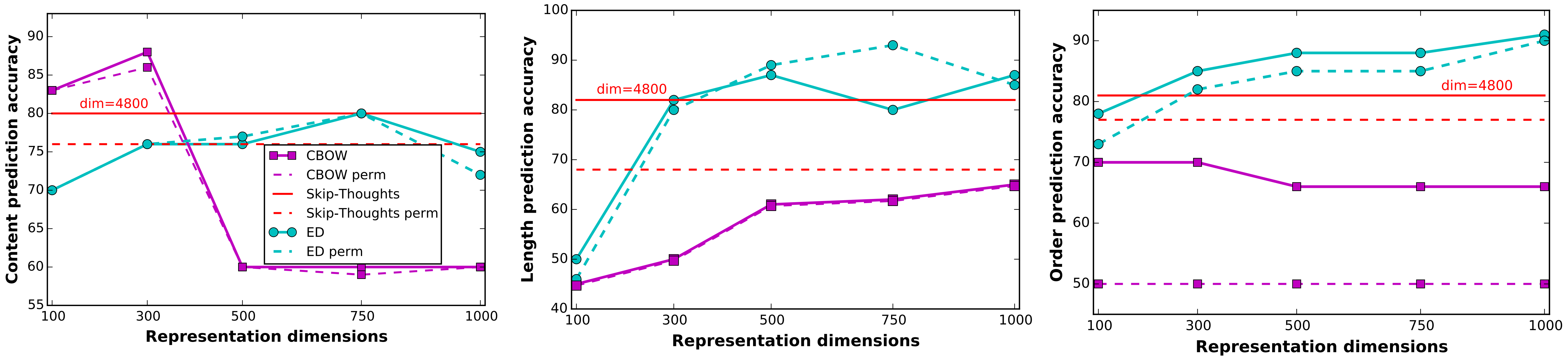


Length



Order

Does it Learn to Represent English or Just Sequences?



Content Length Order

Skip-thought encoders **do care** about the sequence they encode

What did we learn?

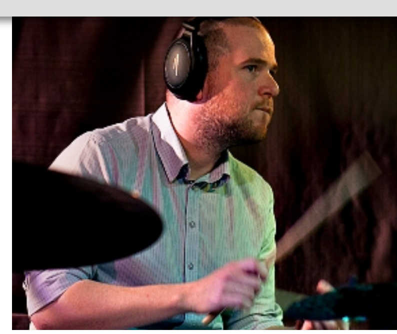
- LSTM-encoder vectors encode length.
- If you care about word identity, prefer CBOW.
- If you care about word order, use LSTM.
- Can recover quite a bit of order also from CBOW.
- LSTM Encoder doesn't rely on language-naturalness
- Skip-thoughts encoder does rely on it.

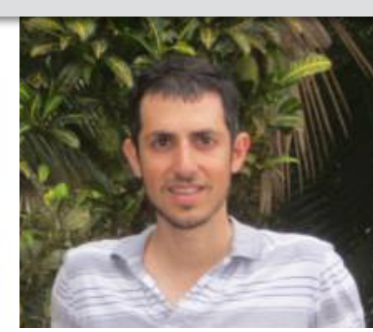
Q2: What is encoded/captured in a vector?

Published as a conference paper at ICLR 2017

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Methodology: can you train a classifier to predict X from the representation?



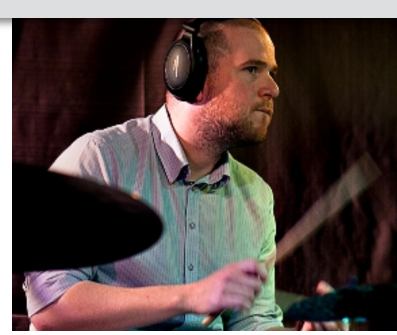


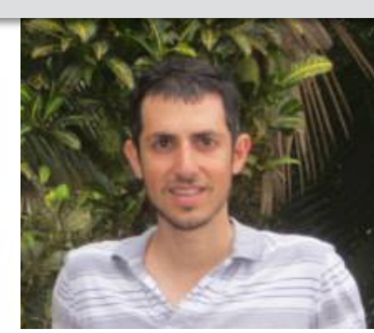
Q2: What is encoded/captured in a vector?

Published as a conference paper at ICLR 2017

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Methodology: can you train a classifier to predict X from the representation?





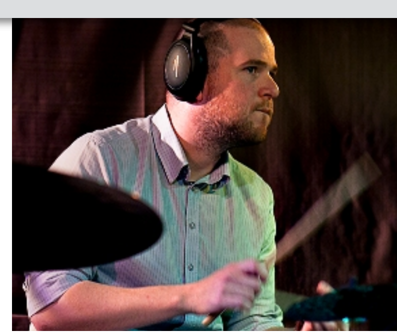
Q2: What is encoded/captured in a vector?

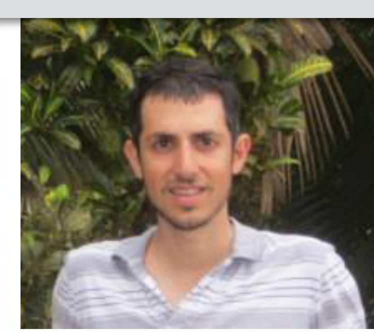
work performed early 2016

Published as a conference paper at ICLR 2017

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Methodology: can you train a classifier to predict X from the representation?





Q2: What is encoded/captured in a vector?

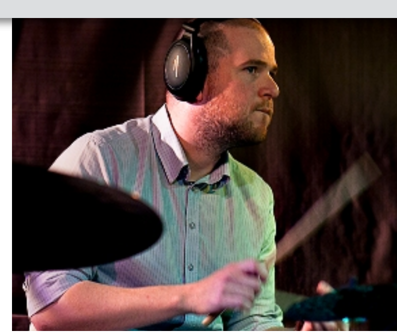
work performed early 2016

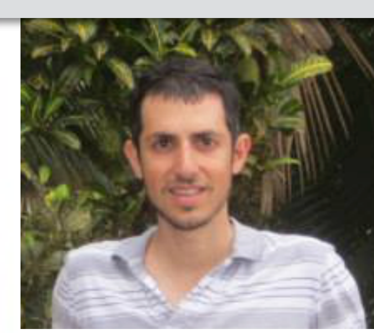
Published as a conference paper at ICLR 2017

Rejected from pretty much all* NLP venues

FINE-GRAINED ANALYSIS OF SENTENCE EMBEDDINGS USING AUXILIARY PREDICTION TASKS

Methodology: can you train a classifier to predict X from the representation?





*that matter

Q2: What is encoded/captured in a vector?

work performed early 2016

Published as a conference paper at ICLR 2017

Rejected from pretty much all* NLP venues

reviewer 2:

```
The paper reads very well, but
a) I do not understand the motivation, and
b) the experiments seem flawed.
```

*that matter

Q2: What is encoded/captured in a vector?

Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure

JAIR

Dieuwke Hupkes D.HUPKES@UVA.NL
Sara Veldhoen S.F.VELDHOEN@UVA.NL
Willem Zuidema ZUIDEMA@UVA.NL
ILLC, University of Amsterdam
P.O.Box 94242,
1090 CE Amsterdam, Netherlands

Q2: What is encoded/captured in a vector?

Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure

JAIR, NIPS workshop 2016

Dieuwke Hupkes D.HUPKES@UVA.NL
Sara Veldhoen S.F.VELDHOEN@UVA.NL
Willem Zuidema ZUIDEMA@UVA.NL
ILLC, University of Amsterdam
P.O.Box 94242,
1090 CE Amsterdam, Netherlands

~with us

Q2: What is encoded/captured in a vector?

much better name!

Visualisation and ‘diagnostic classifiers’ reveal how recurrent and recursive neural networks process hierarchical structure

JAIR, NIPS workshop 2016

Dieuwke Hupkes D.HUPKES@UVA.NL
Sara Veldhoen S.F.VELDHOEN@UVA.NL
Willem Zuidema ZUIDEMA@UVA.NL
ILLC, University of Amsterdam
P.O.Box 94242,
1090 CE Amsterdam, Netherlands

~with us

BIU NLP

Q2: What is encoded/captured in a vector?

Probing for semantic evidence of composition by means of simple classification tasks

RepEval workshop 2016

Allyson Ettinger[1], **Ahmed Elgohary**[2], **Philip Resnik**[1,3]
[1]Linguistics, [2]Computer Science, [3]Institute for Advanced Computer Studies
University of Maryland, College Park, MD
{aetting,resnik}@umd.edu,elgohary@cs.umd.edu

Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure

JAIR, NIPS workshop 2016

Dieuwke Hupkes D.HUPKES@UVA.NL
Sara Veldhoen S.F.VELDHOEN@UVA.NL
Willem Zuidema ZUIDEMA@UVA.NL
ILLC, University of Amsterdam
P.O.Box 94242,
1090 CE Amsterdam, Netherlands

~with us

Q2: What is encoded/captured in a vector?

NIPS 2017

Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems

Yonatan Belinkov and **James Glass**
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{belinkov, glass}@mit.edu

Q2: What is encoded/captured in a vector?

NIPS 2017

Analyzing Hidden Representations in End-to-End Automatic Speech Recognition Systems

IJCNLP 2017

Understanding and Improving Morphological Learning in the Neural Machine Translation Decoder

Fahim Dalvi Nadir Durrani Hassan Sajjad
Yonatan Belinkov* Stephan Vogel

Qatar Computing Research Institute – HBKU, Doha, Qatar
{faimaduddin, ndurrani, hsajjad, svogel}@qf.org.qa

*MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
belinkov@mit.edu

Q2: What is encoded/captured in a vector?

2 years later...

ACL 2018

What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties

Alexis Conneau
Facebook AI Research
Université Le Mans
aconneau@fb.com

German Kruszewski
Facebook AI Research
germank@fb.com

Guillaume Lample
Facebook AI Research
Sorbonne Universités
glample@fb.com

Loïc Barrault
Université Le Mans
loic.barrault@univ-lemans.fr

Marco Baroni
Facebook AI Research
mbaroni@fb.com

Q2: What is encoded/captured in a vector?

2 years later...

ACL 2018

What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties

ACL 2018

Exploring Semantic Properties of Sentence Embeddings

Xunjie Zhu
Rutgers University
Piscataway, NJ, USA
xunjie.zhu@rutgers.edu

Tingfeng Li
Northwestern Polytechnical University, Xi'an, China
ltf@mail.nwpu.edu.cn

Gerard de Melo
Rutgers University
Piscataway, NJ, USA
gdm@demelo.org

Q2: What is encoded/captured in a vector?

2 years later...

ACL 2018 **What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties**

ACL 2018 **Exploring Semantic Properties of Sentence Embeddings**

many more works in xACL / BlackBox NLP

Q2: What is encoded/captured in a vector?

2 years later...

ACL 2018 **What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties**

ACL 2018 **Exploring Semantic Properties of Sentence Embeddings**

many more works in xACL / BlackBox NLP

(ML) workshops --> ML --> non-ACL NLP --> ACL (NAACL, EMNLP...)

is top-tier NLP too conservative?

Q2: What is encoded/captured in a vector?

ACL 2018 **What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties**

ACL 2018 **Exploring Semantic Properties of Sentence Embeddings**

many more works in xACL / BlackBox NLP

(ML) workshops --> ML --> non-ACL NLP --> ACL (NAACL, EMNLP...)

is top-tier NLP too conservative?

You will become reviewers soon. Think about it.

Do I still believe in probing tasks?

- Sort of.
- "BERT network can do SRL with 78%"
  - Useless.
- "BERT network does 78% SRL in layer 3, and 63% in layer 8"
  - Much better.
- They are interesting for **comparing** different networks, **if** we manage to see a difference.
- **But**, hard to interpret the results.

Do I still believe in probing tasks?

- If our classifier managed to extract property X, does this mean the network actually uses property X?

- If our classifier **did not** manage to recover property X, does this mean the network does not use this property?

    - consider: the last layer in a multi-layer network for sentiment, is not predictive of the presence of negation words. Does this mean the network cannot do negation?

Do I still believe in probing tasks?

- Important technique, but take with a grain of salt.

Understanding LSTMs

Q3: what kinds of linguistic structures can be captured by an RNN?

Understanding LSTMs

Q3: what kinds of linguistic structures can be captured by an RNN?

Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies

Tal Linzen[1,2] **Emmanuel Dupoux**[1]
LSCP[1] & IJN[2], CNRS,
EHESS and ENS, PSL Research University
{tal.linzen,
emmanuel.dupoux}@ens.fr

Yoav Goldberg
Computer Science Department
Bar Ilan University
yoav.goldberg@gmail.com

The case for Syntax

- Some natural-language phenomena are indicative of hierarchical structure.
- For example, subject verb agreement.

the boy kicks the ball

the boys kick the ball

The case for Syntax

- Some natural-language phenomena are indicative of hierarchical structure.
- For example, subject verb agreement.

the boy with the white shirt with the blue collar kicks the ball

the boys with the white shirts with the blue collars kick the ball

The case for Syntax

- Some natural-language phenomena are indicative of hierarchical structure.

- For example, subject verb agreement.

the boy (with the white shirt (with the blue collar)) kicks the ball

he boys (with the white shirts (with the blue collars)) kick the ball

The case for Syntax

- Some natural-language phenomena are indicative of hierarchical structure.

- For example, subject verb agreement.

the boy (with the white shirt (with the blue collar)) kicks the ball

he boys (with the white shirts (with the blue collars)) kick the ball

nsubj

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral rationalism are plato and immanuel kant .

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN are plato and immanuel kant .

replace rare words with their POS

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN are plato and immanuel kant .

choose a verb with a subject

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

cut the sentence at the verb

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

plural or singular?

binary prediction task

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

plural or singular?

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

plural or singular?

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

plural or singular?

Can a sequence LSTM learn agreement?

some prominent figures in the history of philosophy who have defended moral NN ____

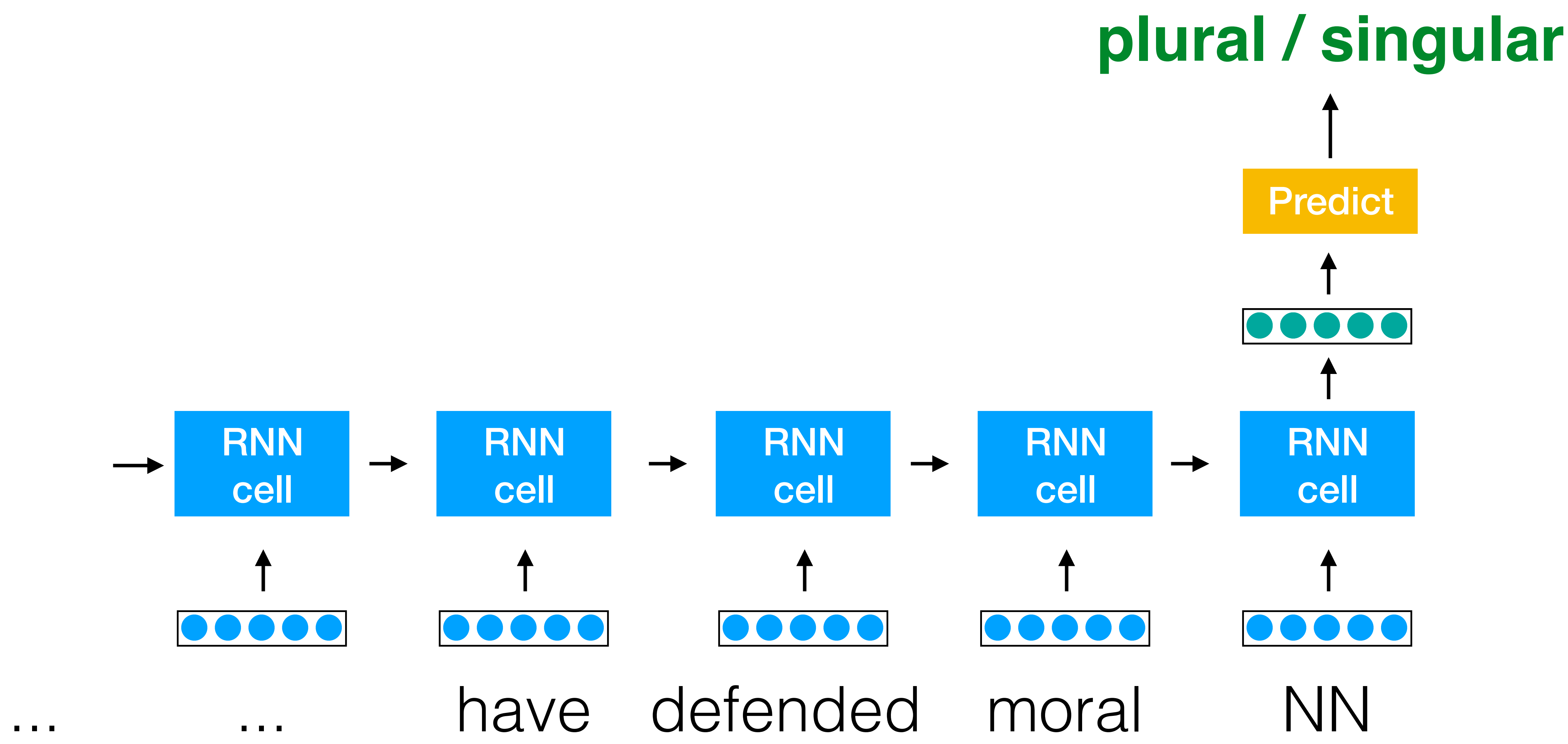
plural or singular?

in order to answer:

Need to learn the concept of number.

Need to identify the **subject** (ignoring irrelevant words)

Binary Prediction Task



Somewhat Harder Task

Somewhat Harder Task

some prominent figures in the history of philosophy who have defended moral NN are plato and immanuel kant .

choose a verb with a subject

Somewhat Harder Task

some prominent figures in the history of philosophy who have defended moral NN are plato and immanuel kant .

some prominent figures in the history of philosophy who have defended moral NN is plato and immanuel kant .

choose a verb with a subject
and flip its number.

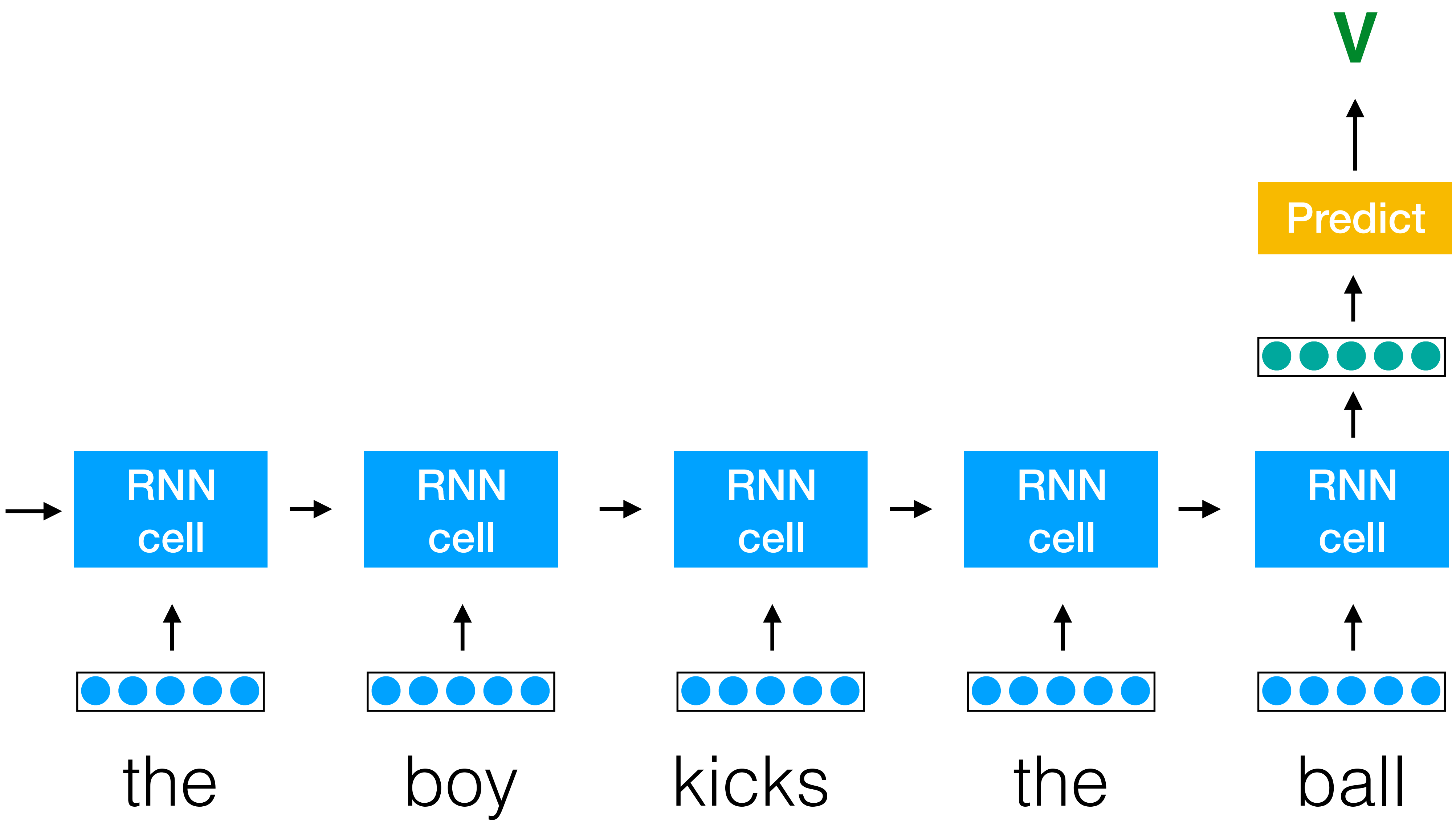
Somewhat Harder Task

some prominent figures in the history of philosophy who have defended moral NN   are plato and immanuel kant . V

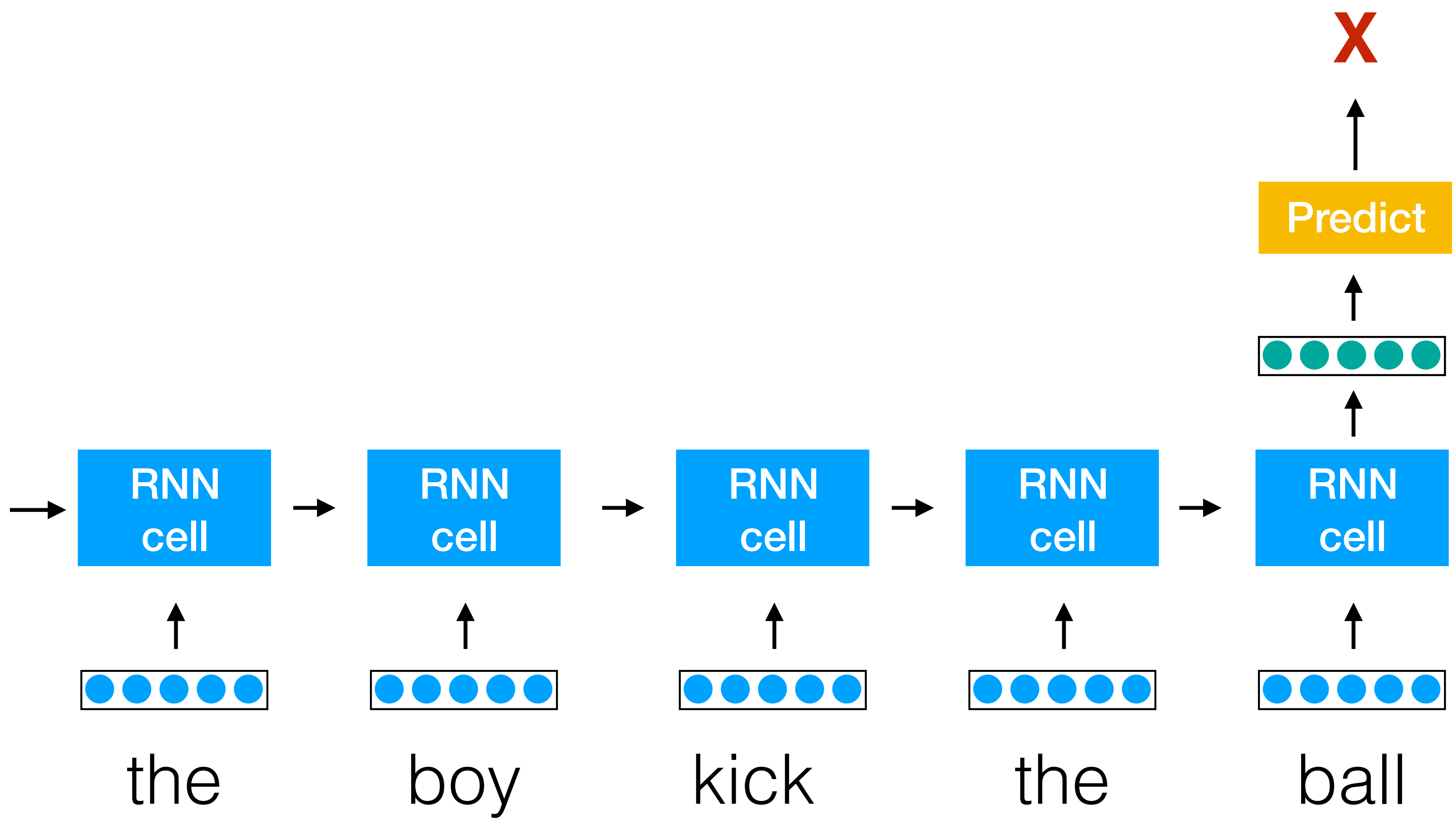
some prominent figures in the history of philosophy who have defended moral NN   is plato and immanuel kant . X

can the LSTM learn to distinguish good from bad sentences?

Sentence Level Task



Sentence Level Task



Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

predicts number with **99**% accuracy.

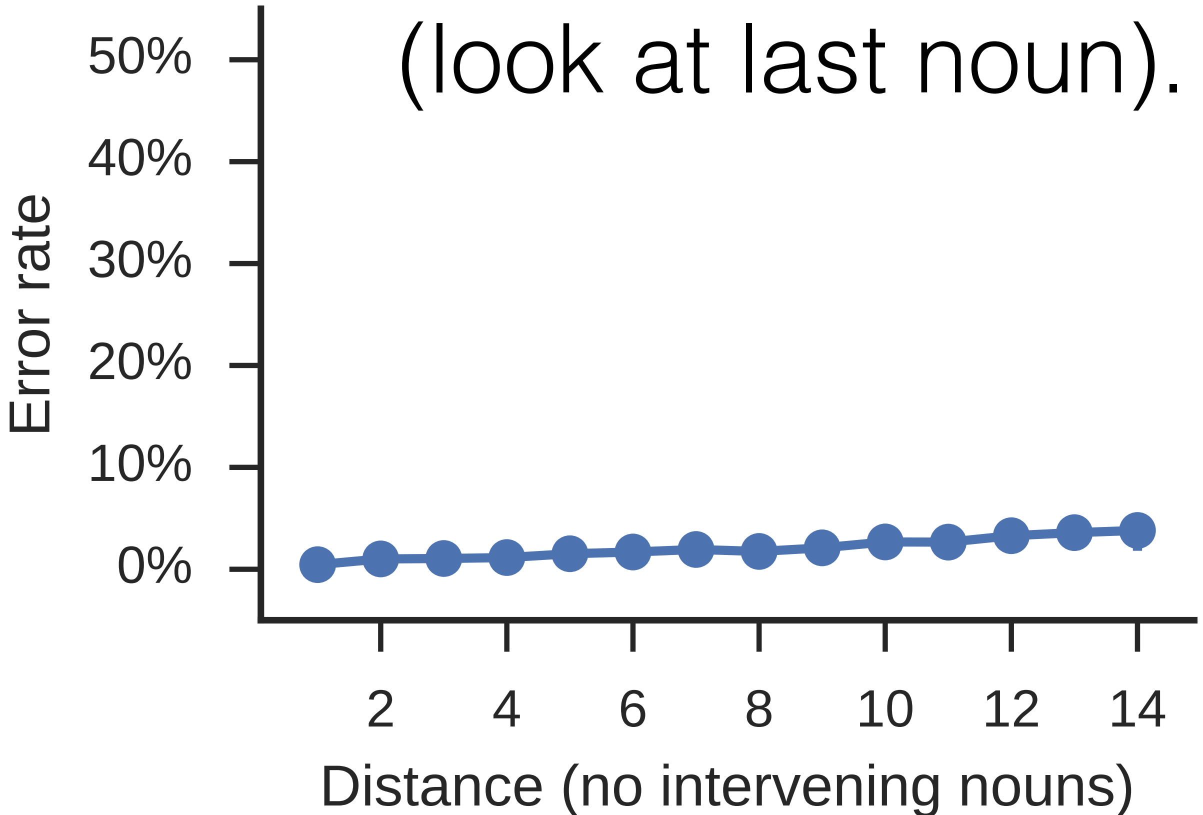
...but most examples are very easy
(look at last noun).

Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

predicts number with **99**% accuracy.

...but most examples are very easy (look at last noun).



Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

predicts number with **99**% accuracy.

...but most examples are very easy
(look at last noun).

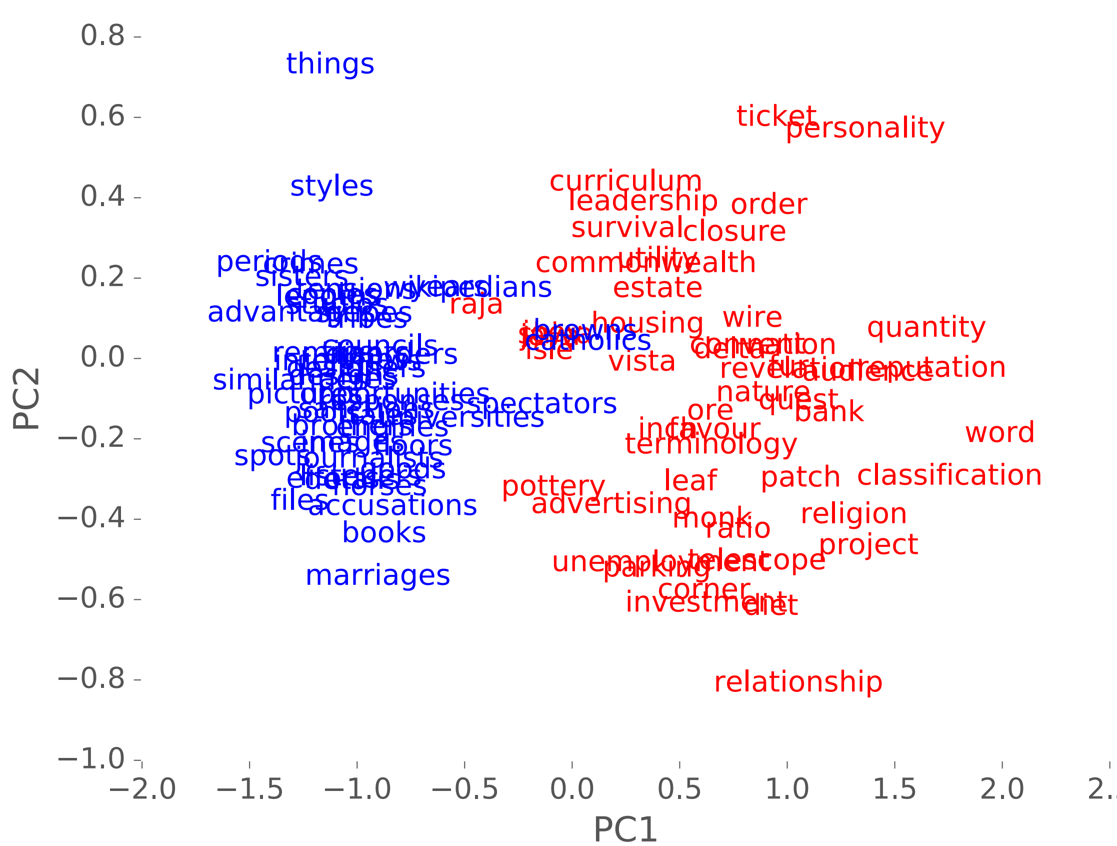
when restricted to cases
of at least one intervening noun:

97% accuracy

Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

learns number of nouns

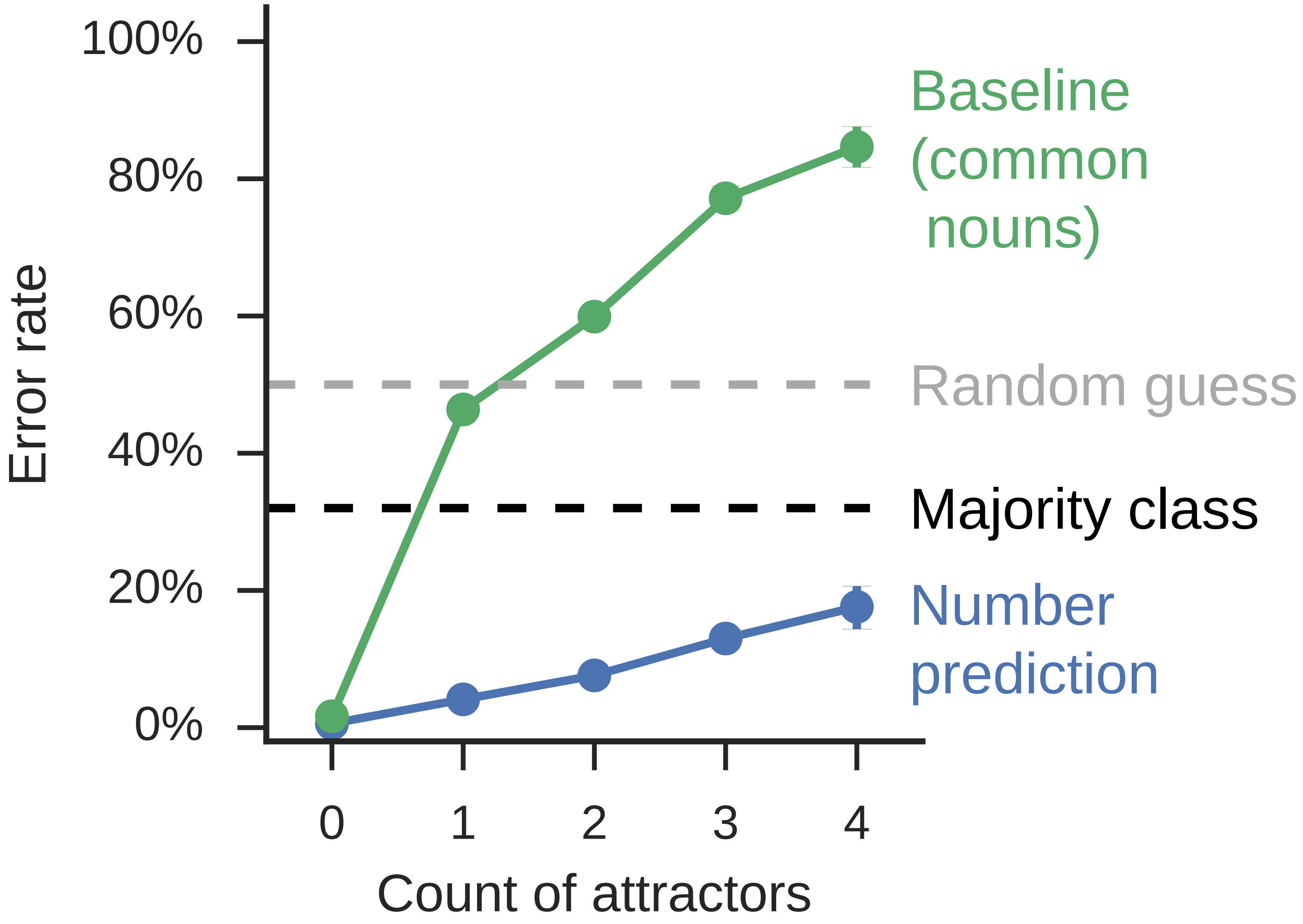


Can a sequence LSTM learn agreement?

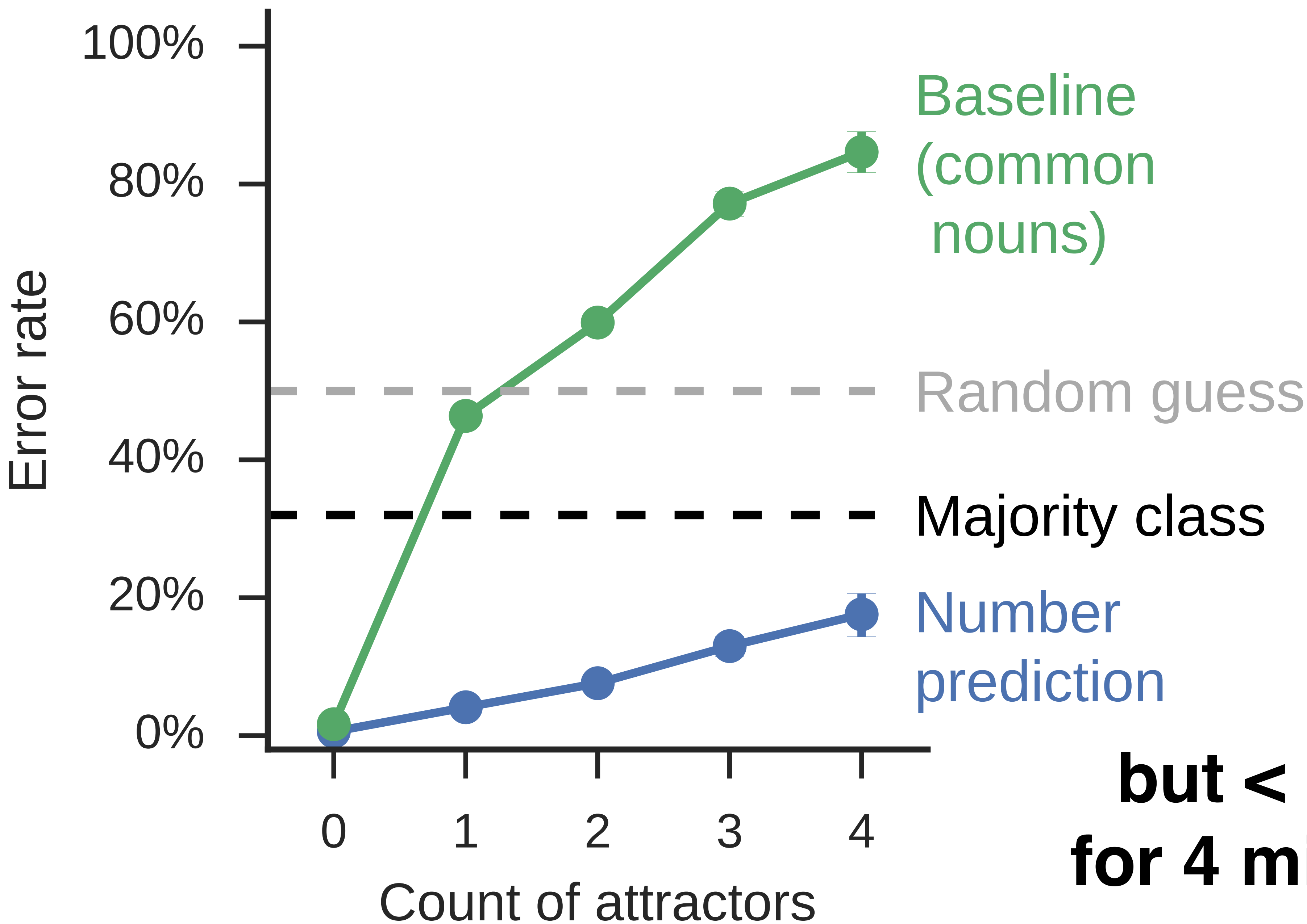
LSTMs learn agreement remarkably well.

more errors as the number of intervening nouns of opposite number increases

Can a sequence LSTM learn agreement?



Can a sequence LSTM learn agreement?



but < 16% err for 4 misleading nouns...

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

in many and diverse cases.

but we did manage to find some common trends.

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

noun compounds can be tricky

Conservation **refugees live** in a world colored in shades of gray; limbo.

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

Relative clauses are hard.

The **landmarks** *that* this article lists here **are** also run-of-the-mill and not notable.

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

Reduced relative clauses are harder.

The **landmarks** this article lists here **are** also run-of-the-mill and not notable.

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

	Error
No relative clause	3.2%
Overt relative clause	9.9%
Reduced Relative clause	**25%**

Can a sequence LSTM learn agreement?

Where do LSTMs fail?

	Error
No relative clause	3.2%
Overt relative clause	9.9%
Reduced Relative clause	**25%**

humans also fail much more on reduced relatives.

The agreement experiment: recap

- We wanted to show LSTMs can't learn hierarchy.
    - **--> We sort-of failed.**
- **LSTMs learn to cope with natural-language patterns that exhibit hierarchy, based on minimal and indirect supervision.**
- But some sort of relevant supervision is required.

Can a Transformer Learn agreement?

Assessing BERT's Syntactic Abilities

Yoav Goldberg[1,2]
[1] Computer Science Department, Bar Ilan University
[2] Allen Institute for Artificial Intelligence
yogo@cs.biu.ac.il , yoav@allenai.org

Can a Transformer Learn agreement?

Attractors	BERT Base	BERT Large	# sents
1	0.97	0.97	24031
2	0.97	0.97	4414
3	0.96	0.96	946
4	0.97	0.96	254

BERT does extremely well

Assessing BERT's Syntactic Abilities

Yoav Goldberg[1,2]
[1] Computer Science Department, Bar Ilan University
[2] Allen Institute for Artificial Intelligence
yogo@cs.biu.ac.il , yoav@allenai.org

The agreement experiment: recap

- I wanted to show Transformers can't learn hierarchy.
  - **--> Major fail. They are amazing.**
    - **But how do they do it??**

we don't know. :-(
yet.

The agreement experiment: aftermath

This triggered **a lot** of very interesting work!

Colorless green recurrent networks dream hierarchically

Kristina Gulordava*
Department of Linguistics
University of Geneva
kristina.gulordava@unige.ch

Piotr Bojanowski
Facebook AI Research
Paris
bojanowski@fb.com

Edouard Grave
Facebook AI Research
New York
egrave@fb.com

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

Marco Baroni
Facebook AI Research
Paris
mbaroni@fb.com

The agreement experiment: aftermath

This triggered **a lot** of very interesting work!

Colorless green recurrent networks dream hierarchically

Kristina Gulordava[*]
Department of Linguistics
University of Geneva
kristina.gulordava@unige.ch

Piotr Bojanowski
Facebook AI Research
Paris
bojanowski@fb.com

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

F

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better

Adhiguna Kuncoro[♠♣] **Chris Dyer**[♠] **John Hale**[♠♡]
Dani Yogatama[♠] **Stephen Clark**[♠] **Phil Blunsom**[♠♣]
[♠]DeepMind, London, UK
[♣]Department of Computer Science, University of Oxford, UK
[♡]Department of Linguistics, Cornell University, NY, USA
{akuncoro,cdyer,jthale,dyogatama,clarkstephen,pblunsom}@google.com

The agreement experiment: aftermath

This triggered **a lot** of very interesting work!

Colorless green recurrent networks dream hierarchically

Kristina Gulordava*
Department of Linguistics
University of Geneva
kristina.gulordava@unige.ch

Piotr Bojanowski
Facebook AI Researcl
Paris
bojanowski@fb.com

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better

Chris Dyer♠ John Hale♠♡
ɔhen Clark♠ Phil Blunsom♠♣
ind, London, UK
Science, University of Oxford, UK
ics, Cornell University, NY, USA
ıma,clarkstephen,pblunsom}@google.com

Targeted Syntactic Evaluation of Language Models

Rebecca Marvin
Department of Computer Science
Johns Hopkins University
becky@jhu.edu

Tal Linzen
Department of Cognitive Science
Johns Hopkins University
tal.linzen@jhu.edu

The agreement experiment: aftermath

This triggered **a lot** of very interesting work!

Colorless green recurrent networks dream hierarchically

Kristina Gulordava*
Department of Linguistics
University of Geneva
kristina.gulordava@unige.ch

Piotr Bojanowski
Facebook AI Researcl
Paris
bojanowski@fb.com

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better

Chris Dyer♠ John Hale♠♡
hen Clark♠ Phil Blunsom♠♣

Targeted Syntactic Evaluation of Language Models

Rebecca Marvin
Department of Computer Science
Johns Hopkins University
becky@jhu.edu

Depart
Joh
tal

RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency

Richard Futrell[1], **Ethan Wilcox**[2], **Takashi Morita**[3,4], and **Roger Levy**[5]

[1]Department of Language Science, UC Irvine, rfutrell@uci.edu
[2]Department of Linguistics, Harvard University, wilcoxeg@g.harvard.edu
[3]Primate Research Institute, Kyoto University, tmorita@alum.mit.edu
[4]Department of Linguistics and Philosophy, MIT
[5]Department of Brain and Cognitive Sciences, MIT, rplevy@mit.edu

The agreement experiment: aftermath

This triggered **a lot** of very interesting work!

Colorless green recurrent networks dream hierarchically

Kristina Gulordava*
Department of Linguistics
University of Geneva
kristina.gulordava@unige.ch

Piotr Bojanowski
Facebook AI Research
Paris
bojanowski@fb.com

LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better

Chris Dyer♠ John Hale♠♡
hen Clark♠ Phil Blunsom♠♣

Targeted Syntactic Evaluation of Language Models

Rebecca Marvin
Department of Computer Science
Johns Hopkins University
becky@jhu.edu

Depart
Joh

RNNs as psycholinguistic subjects: Syntactic state and grammatical dependency

Richard Futrell[1], **Ethan Wilcox**[2], **Takashi Morita**[3,4], and **Roger Levy**[5]

[1]Department of Language Science, UC Irvine, rfutrell@uci.edu
it of Linguistics, Harvard University, wilcoxeg@g.harvard.edu
Research Institute, Kyoto University, tmorita@alum.mit.edu
[4]Department of Linguistics and Philosophy, MIT
[5]Department of Brain and Cognitive Sciences, MIT, rplevy@mit.edu

many others

English is so simple though.
Let's crank up the complexity.

English is so simple though. Let's crank up the complexity.

Can LSTM Learn to Capture Agreement? The Case of Basque

Shauli Ravfogel, Francis M. Tyers, Yoav Goldberg

(Submitted on 11 Sep 2018 (v1), last revised 26 Nov 2018 (this version, v4))

Basque is **complex**

- Verbs agree with ***all*** their arguments (polypersonal agreement)
- **Explicit case marking** on NPs
- Relatively **flexible word order**
- **Ergative** case system
- Morphologically **rich**

Basque is **complex**

All scores in Basque were much lower than in English.

Basque is **complex**

All scores in Basque were much lower than in English.

But why?

Limited data? Poly-personal agreement? Ergativity?
Word-order? Different domains?

The science way:

Better variable control

BIU NLP

The science way:

Better variable control

Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages

Shauli Ravfogel, Yoav Goldberg, Tal Linzen

(Submitted on 15 Mar 2019 (v1), last revised 26 Mar 2019 (this version, v2))

AI2 ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

The science way:

Better variable control

Studying the Inductive Biases of RNNs with Synthetic Variations of Natural Languages

Shauli Ravfogel, Yoav Goldberg, Tal Linzen

(Submitted on 15 Mar 2019 (v1), last revised 26 Mar 2019 (this version, v2))

Create synthetic variants of English corpus imitating the phenomena we care about

English + Polypersonal Agreement

they say*kon* the broker took*kar**ker* them out for lunch frequently .

(*kon*: *plural subject*; *kar*: *singular subject*; *ker*: *plural object*)


English ~> Word Orders

SVO	they say the broker took out frequently them for lunch .
SOV	they the broker them took out frequently for lunch say .
VOS	say took out frequently them the broker for lunch they.
VSO	say they took out frequently the broker them for lunch .
OSV	them the broker took out frequently for lunch they say .
OVS	them took out frequently the broker for lunch say they .

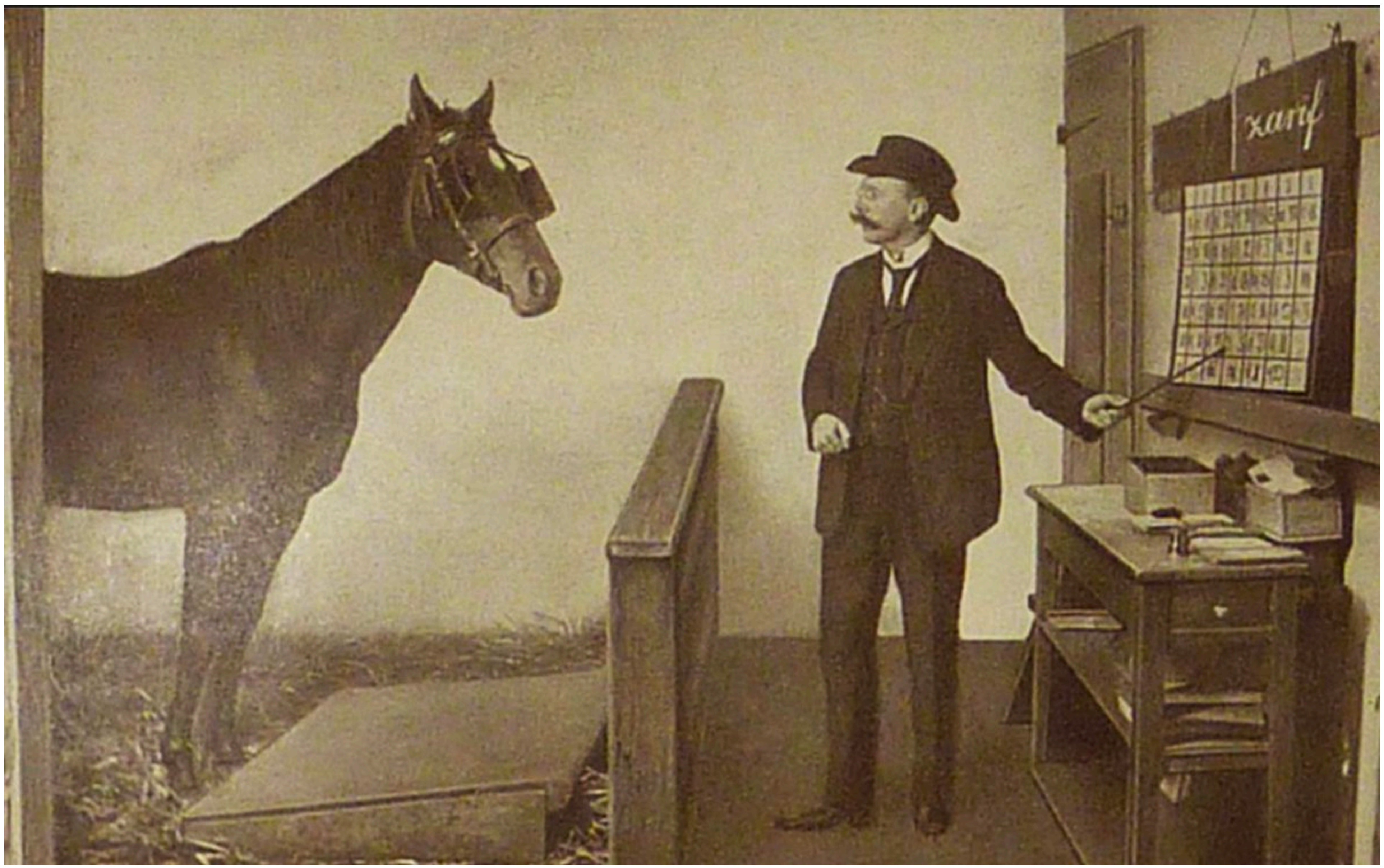

English + Case Marking

Unambiguous	theykon saykon the brokerkar tookkarker theyker out for lunch frequently .
	(kon: plural subject; kar: singular subject; ker: plural object)
Syncretic	theykon saykon the brokerkar tookkarkar theykar out for lunch frequently .
	(kon: plural subject; kar: plural object/singular subject)
Argument marking	theyker sayker the brokerkin tookkerkin theyker out for lunch frequently .
	(*ker: plural argument; kin: singular argument*)

Conclusions? Check the paper.

Q4: when do models fail? what did they *really* learn?

Q4: when do models fail? what did they *really* learn?



google 'clever hans'

This horse knows how to perform math!!

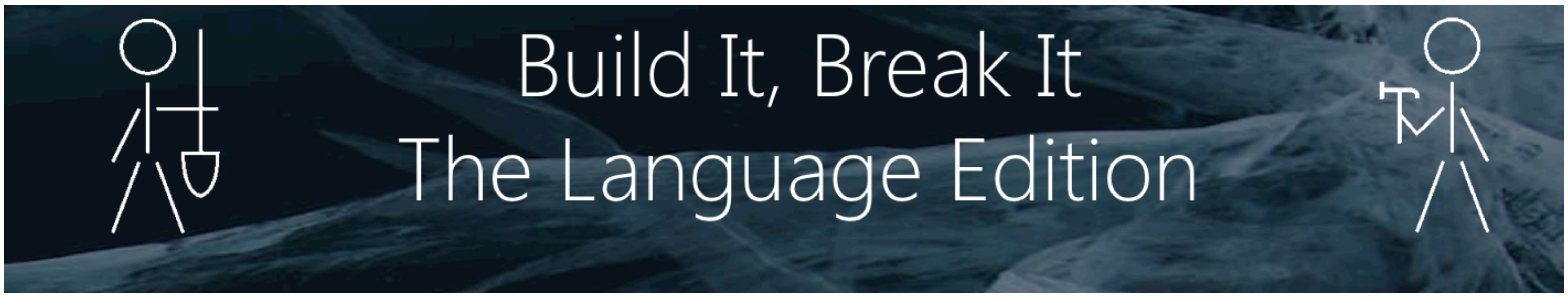
Q4: when do models fail? what did they *really* learn?

Methodology:
create *specific examples* that
make seemingly great models *fail*.

Methodology:

create *specific examples* that
make seemingly great models *fail*.

Q4: when do models fail? what did they *really* learn?



Methodology:

create *specific examples* that
make seemingly great models *fail*.

Q4: when do models fail? what did they *really* learn?

ACL 2018

Breaking NLI Systems with Sentences that Require Simple Lexical Inferences

Max Glockner[1], Vered Shwartz[2] and Yoav Goldberg[2]

[1]Computer Science Department, TU Darmstadt, Germany
[2]Computer Science Department, Bar-Ilan University, Ramat-Gan, Israel
{maxg216,vered1986,yoav.goldberg}@gmail.com

Methodology:
create *specific examples* that
make seemingly great models *fail*.

Q4: when do models fail? what did they *really* learn?

ACL 2019

Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference

R. Thomas McCoy,[1] Ellie Pavlick,[2] & Tal Linzen[1]
[1]Department of Cognitive Science, Johns Hopkins University
[2]Department of Computer Science, Brown University
tom.mccoy@jhu.edu, ellie_pavlick@brown.edu, tal.linzen@jhu.edu

Methodology:

create *specific examples* that
make seemingly great models *fail*.

Q4: when do models fail? what did they *really* learn?

Heuristic	Definition	Example
Lexical overlap	Assume that a premise entails all hypotheses constructed from words in the premise	**The doctor** was **paid** by **the actor**. $\xrightarrow[\text{WRONG}]{}$ The doctor paid the actor.
Subsequence	Assume that a premise entails all of its contiguous subsequences.	The doctor near **the actor danced**. $\xrightarrow[\text{WRONG}]{}$ The actor danced.
Constituent	Assume that a premise entails all complete subtrees in its parse tree.	If **the artist slept**, the actor ran. $\xrightarrow[\text{WRONG}]{}$ The artist slept.

Table 1: The heuristics targeted by the HANS dataset, along with examples of incorrect entailment predictions that these heuristics would lead to.

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what did they *really* learn?

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what did they *really* learn?

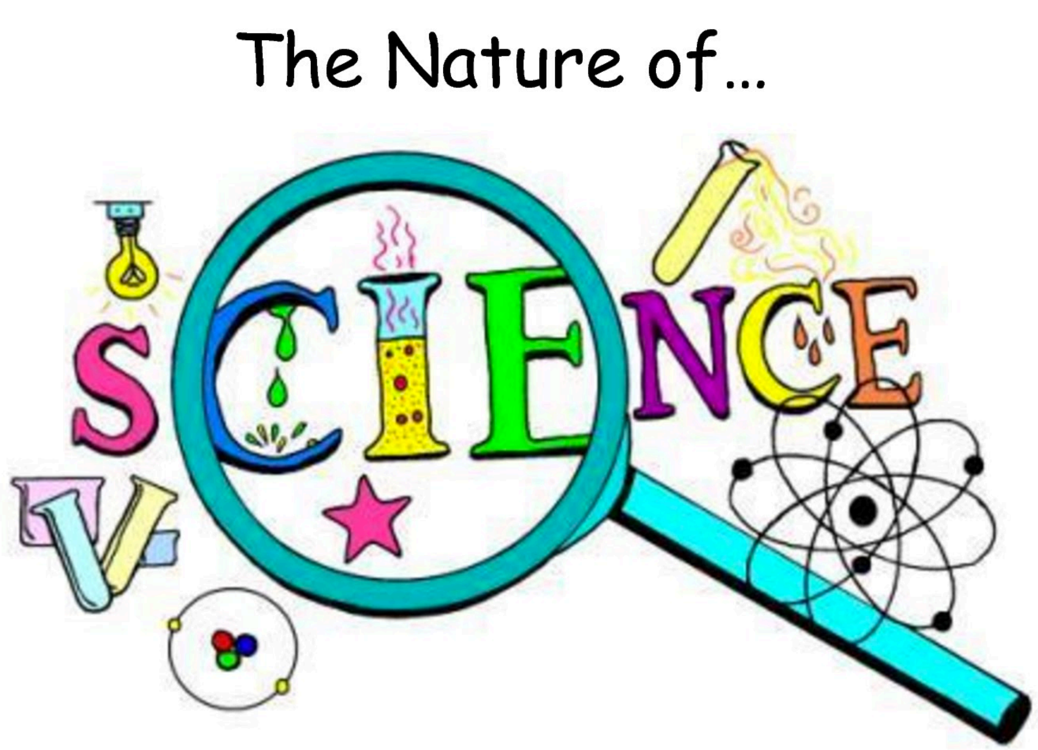
The Nature of...

Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what did they *really* learn?



Treat the representations / model as an "organism".

Come up with hypotheses.
Perform experiments.

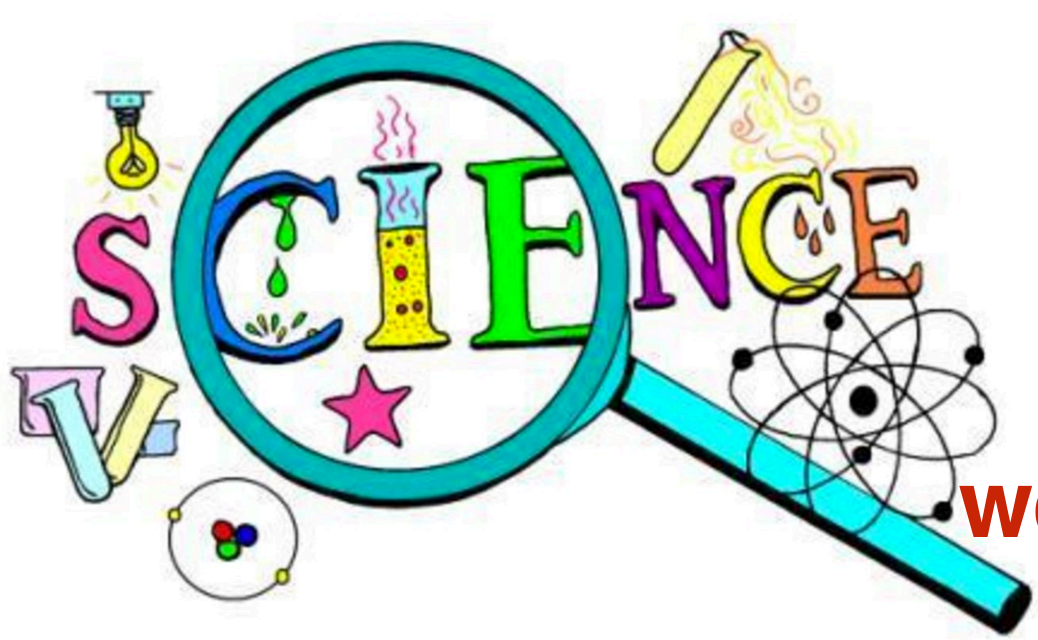
Q1: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q2: What is encoded/captured in a vector?

Q3: what kinds of linguistic structures
can be captured by an RNN?

Q4: when do models fail? what did they *really* learn?

The Nature of...



Treat the representations / model as an "organism".

Come up with hypotheses.
Perform experiments.

we never learned to do this in CS :(

our latest effort in this space (no slides yet..)

oLMpics - On what Language Model Pre-training Captures

Alon Talmor[1,2] Yanai Elazar[1,3] Yoav Goldberg[1,3] Jonathan Berant[1,2]

[1]The Allen Institute for AI
[2]Tel-Aviv University
[3]Bar-Ilan University
{alontalmor@mail,joberant@cs}.tau.ac.il,
{yanaiela,yoav.goldberg}@gmail.com

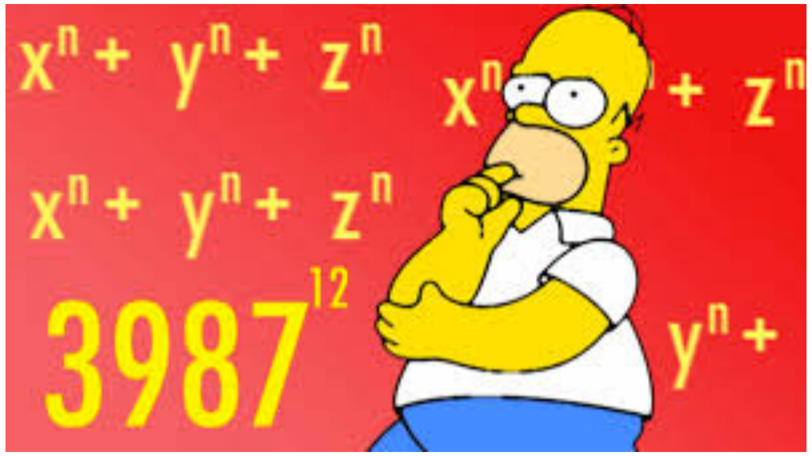
Q4: when do models fail? what did they *really* learn?

Q5: What is the representation power of different architectures?

Q6: Extracting a discrete representation from a trained model.

Q5: What is the representation power of different architectures?

Q6: Extracting a discrete representation from a trained model.

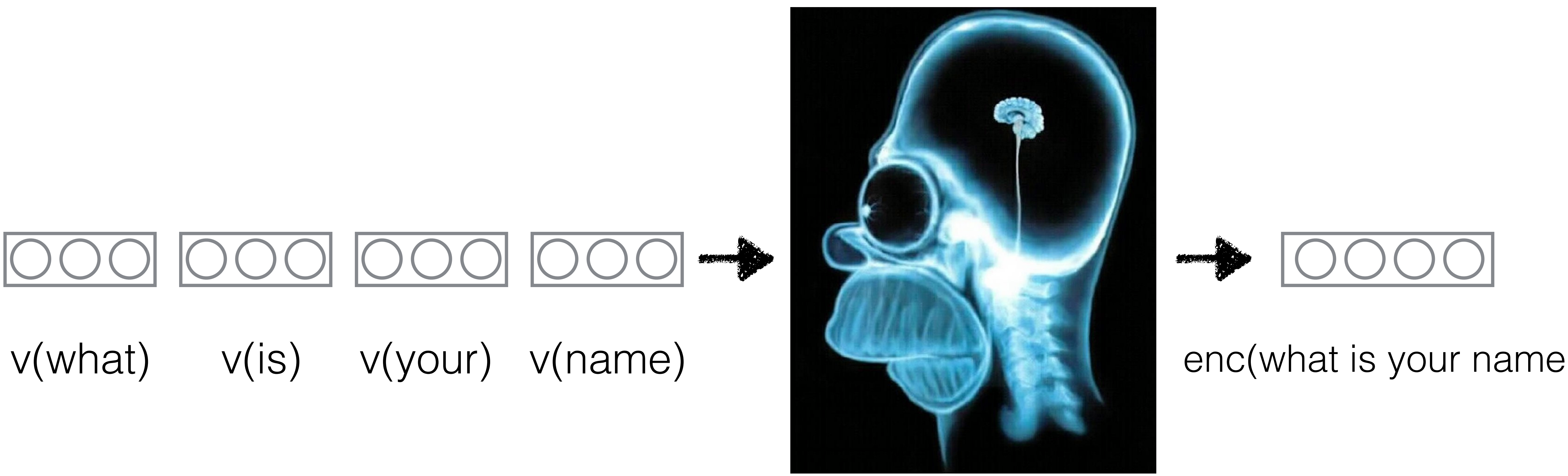


Back to a "familiar territory". Computer science. Math.

Agenda

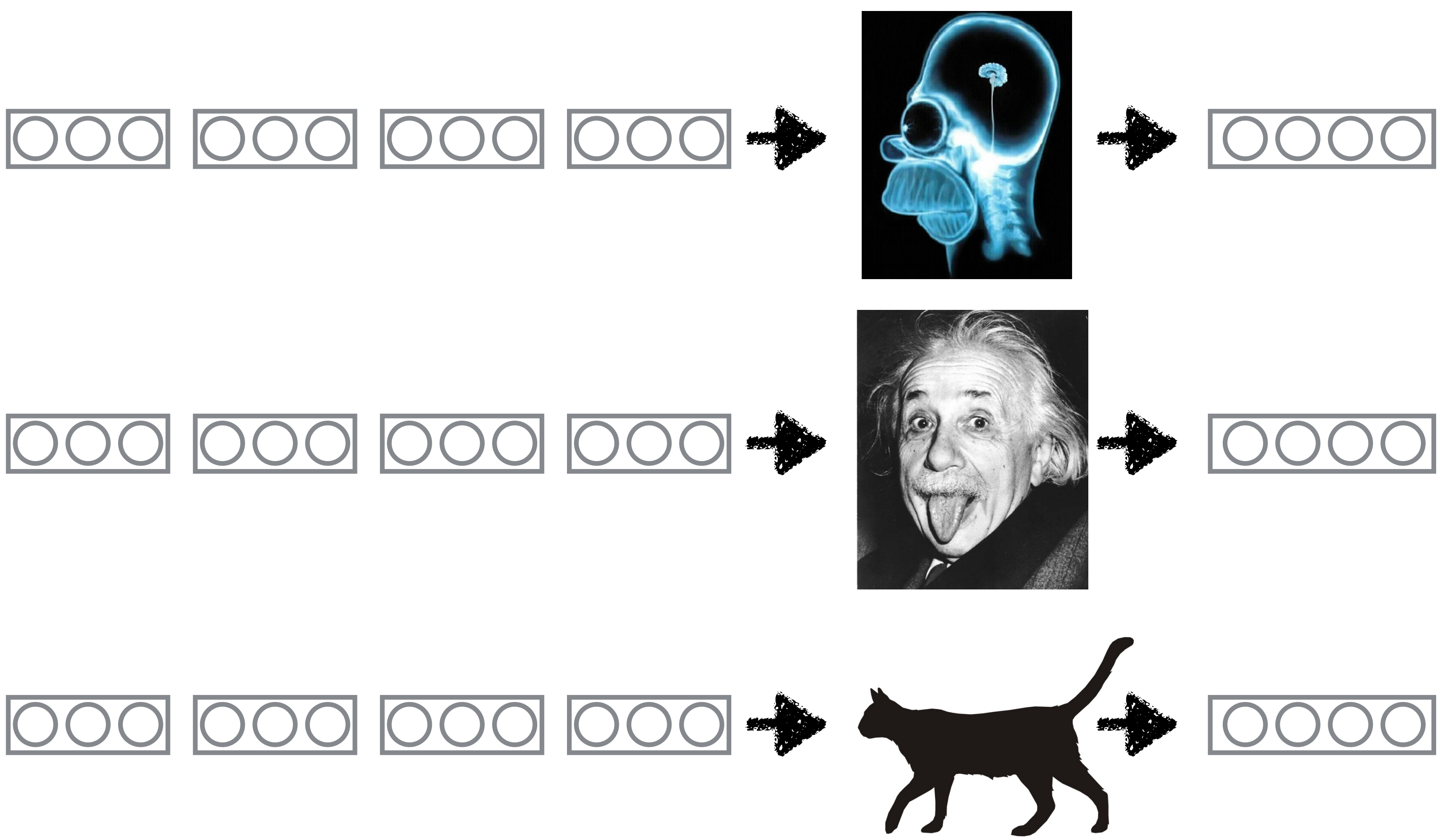
- RNNs
- Formal expressive power of RNNs
- Extracting FSAs from RNNs

Recurrent Neural Networks



- Very strong models of sequential data.
- **Trainable** function from *n* vectors to a single vector.

Recurrent Neural Networks



- There are different variants (implementations).
- Same interface. Same power?

Q5: What is the representation power of different architectures?

Q5: What is the representation power of different architectures?

Recurrent Neural Networks as Weighted Language Recognizers

Yining Chen
Dartmouth College
yining.chen.18@dartmouth.edu

Sorcha Gilroy
ILCC
University of Edinburgh
s.gilroy@sms.ed.ac.uk

Andreas Maletti
Institute of Computer Science
Universität Leipzig
andreas.maletti@uni-leipzig.de

Jonathan May
Information Sciences Institute
University of Southern California
jonmay@isi.edu

Kevin Knight
Information Sciences Institute
University of Southern California
knight@isi.edu

Rational Recurrences

Hao Peng[◊] **Roy Schwartz**[◊♡] **Sam Thomson**[♣] **Noah A. Smith**[◊♡]
[◊]Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA
[♣]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[♡]Allen Institute for Artificial Intelligence, Seattle, WA, USA
{hapeng,roysch,nasmith}@cs.washington.edu, sthomson@cs.cmu.edu

Q5: What is the representation power of different architectures?

Recurrent Neural Networks as Weighted Language Recognizers

Yining Chen
Dartmouth College
yining.chen.18@dartmouth.edu

Sorcha Gilroy
ILCC
University of Edinburgh
s.gilroy@sms.ed.ac.uk

Andreas Maletti
Institute of Computer Science
Universität Leipzig
andreas.maletti@uni-leipzig.de

Jonathan May
Information Sciences Institute
University of Southern California
jonmay@isi.edu

Kevin Knight
Information Sciences Institute
University of Southern California
knight@isi.edu

Q5: What is the representation power of different architectures?

are all RNNs equivalent?

On the Practical Computational Power of Finite Precision RNNs for Language Recognition

Gail Weiss
Technion, Israel

Yoav Goldberg
Bar-Ilan University, Israel

Eran Yahav
Technion, Israel

{sgailw,yahave}@cs.technion.ac.il
yogo@cs.biu.ac.il

RNNs have Turing Power?

RNNs have Turing Power?

On the Computational Power of Neural Nets*

HAVA T. SIEGELMANN†

Department of Information Systems Engineering, Technion, Haifa 32000, Israel

AND

EDUARDO D. SONTAG‡

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903

Received February 4, 1992; revised May 24, 1993

YES, THEY DO!

RNNs have Turing Power?

On the Computational Power of Neural Nets*

HAVA T. SIEGELMANN†

Department of Information Systems Engineering, Technion, Haifa 32000, Israel

AND

EDUARDO D. SONTAG‡

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903

Received February 4, 1992; revised May 24, 1993

YES, THEY DO!

But this answer is not very useful.

RNNs have Turing Power?

On the Computational Power of Neural Nets*

Proof requires infinite precision.
"push 0 into stack": g = g/4 + 1/4
this allows pushing **15** zeros when using 32 bit floating point.

Department of Mathematics, Rutgers University, New Brunswick, New Jersey 08903

Received February 4, 1992; revised May 24, 1993

YES, THEY DO!

But this answer is not very useful.

RNNs have Turing Power?

On the Computational Power of Neural Nets*

Construction requires complex combination of many carefully crafted components.

can this really be reached by gradient methods?

Received February 4, 1992; revised May 24, 1993

YES, THEY DO!

But this answer is not very useful.

RNNs have Turing Power?

On the Computational Power of Neural Nets*

Construction requires extra processing time at the end of the sequence.

we use "real time" RNNs in practice.

Received February 4, 1992; revised May 24, 1993

YES, THEY DO!

But this answer is not very useful.

RNN Flavors

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

"Classic" RNNs

Elman RNN (SRNN) Saturating activation.

$$h_t = \tanh(Wx_t + Uh_{t-1} + b)$$

IRNN ReLU activation.

$$h_t = max(0, (Wx_t + Uh_{t-1} + b))$$

RNN Flavors

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

Gated RNNs

Gated Recurrent Unit

$$\begin{aligned}
z_t &= \sigma(W^z x_t + U^z h_{t-1} + b^z) \\
r_t &= \sigma(W^r x_t + U^r h_{t-1} + b^r) \\
\tilde{h}_t &= \tanh(W^h x_t + U^h (r_t \circ h_{t-1}) + b^h) \\
h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t
\end{aligned}$$

LSTM

$$\begin{aligned}
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\
\tilde{c}_t &= \tanh(W^c x_t + U^c h_{t-1} + b^c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ g(c_t)
\end{aligned}$$

RNN Flavors

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

With finite precision, Elman RNNs are **Finite State**.

We do not know much about other flavors.

Common Wisdom

Gated architectures (GRU, LSTM)
are better than
non-Gated architectures (SRNN, IRNN)

~~Common Wisdom~~

Gated architectures (GRU, LSTM)
are better than
non-Gated architectures (SRNN, IRNN)

we show that in terms of **expressive power,**
there is an aspect in which:

LSTM > GRU
IRNN > SRNN

Power of Counting

Counter Machines and Counter Languages*,†

by

PATRICK C. FISCHER‡
Cornell University
Ithaca, New York

and

ALBERT R. MEYER¶ and ARNOLD L. ROSENBERG
IBM Watson Research Center
Yorktown Heights, New York

(1968)

Power of Counting

Counter Machines and Counter Languages*,†

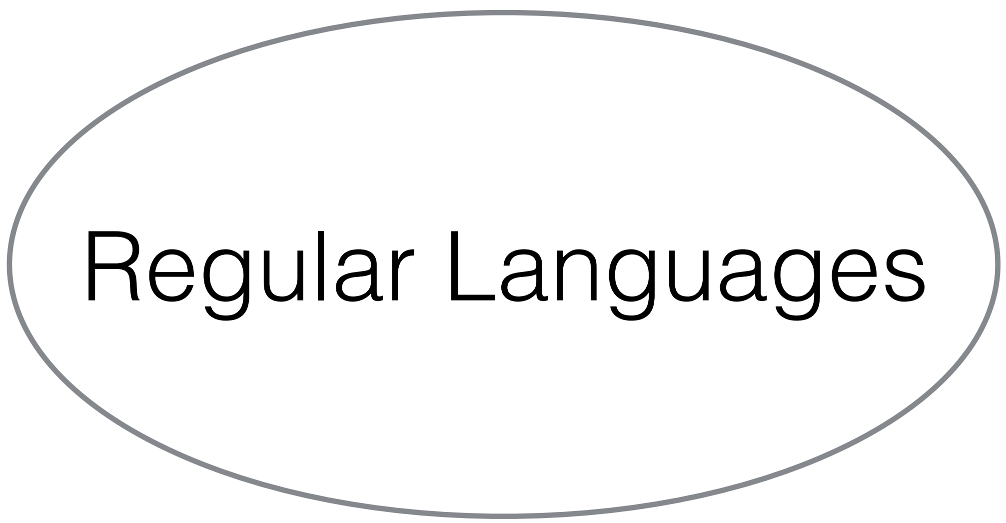
counter machines are
Finite State Automata with k counters.

INC, DEC, Compare0

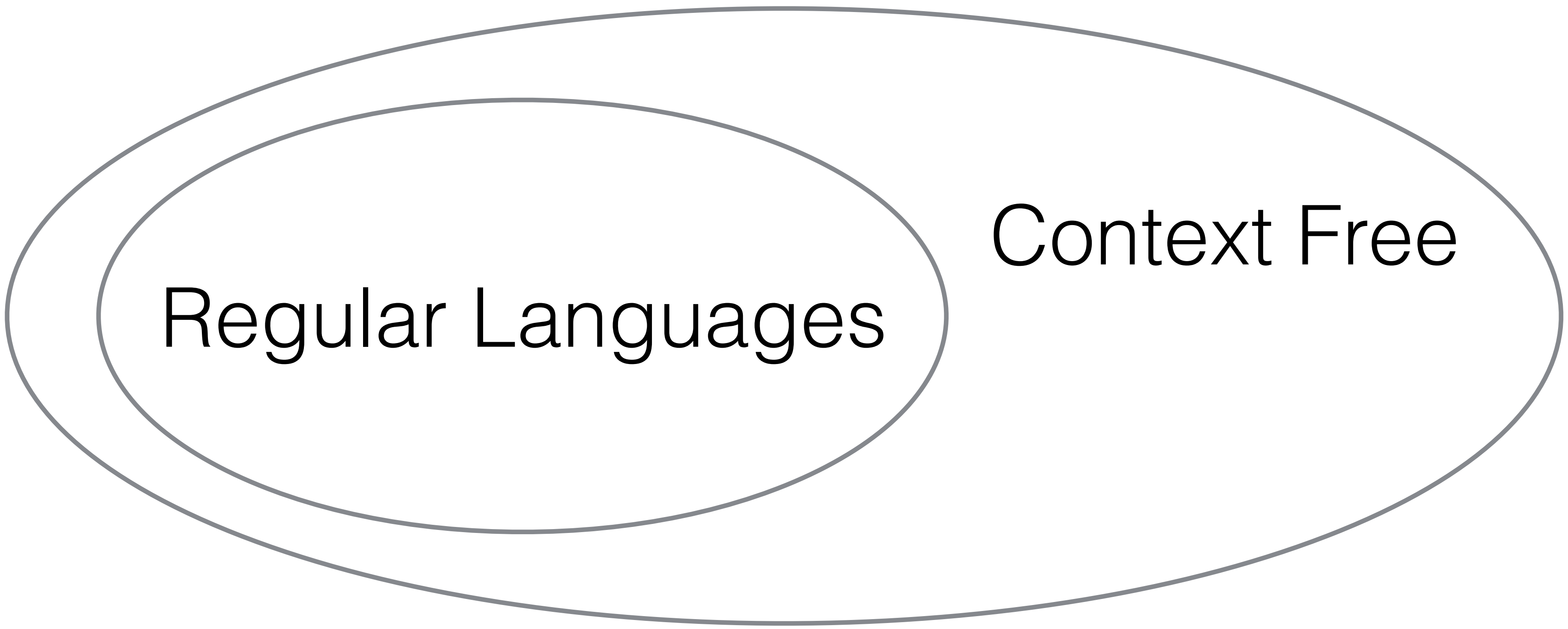
Yorktown Heights, New York

(1968)

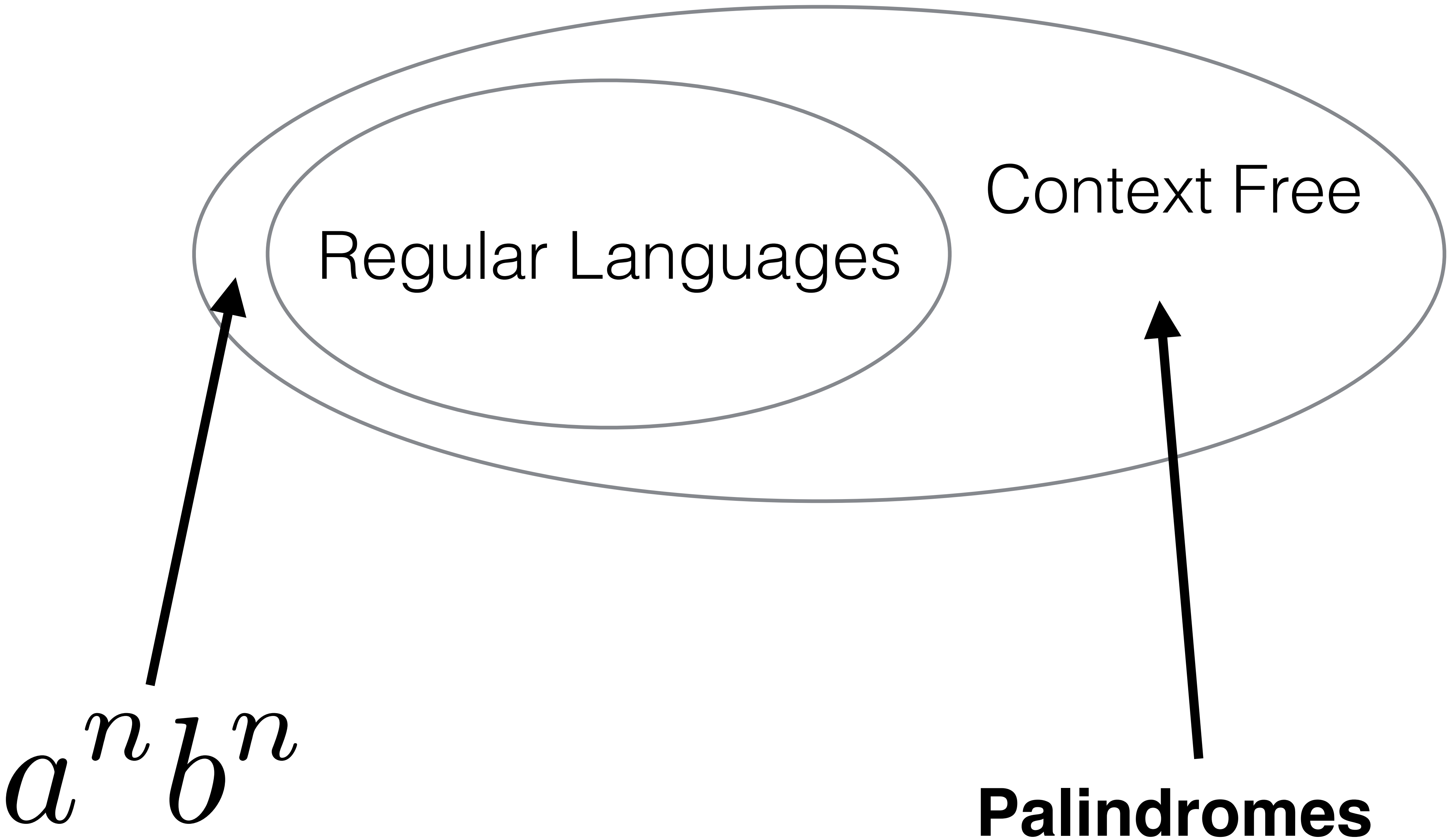
Chomsky Hierarchy



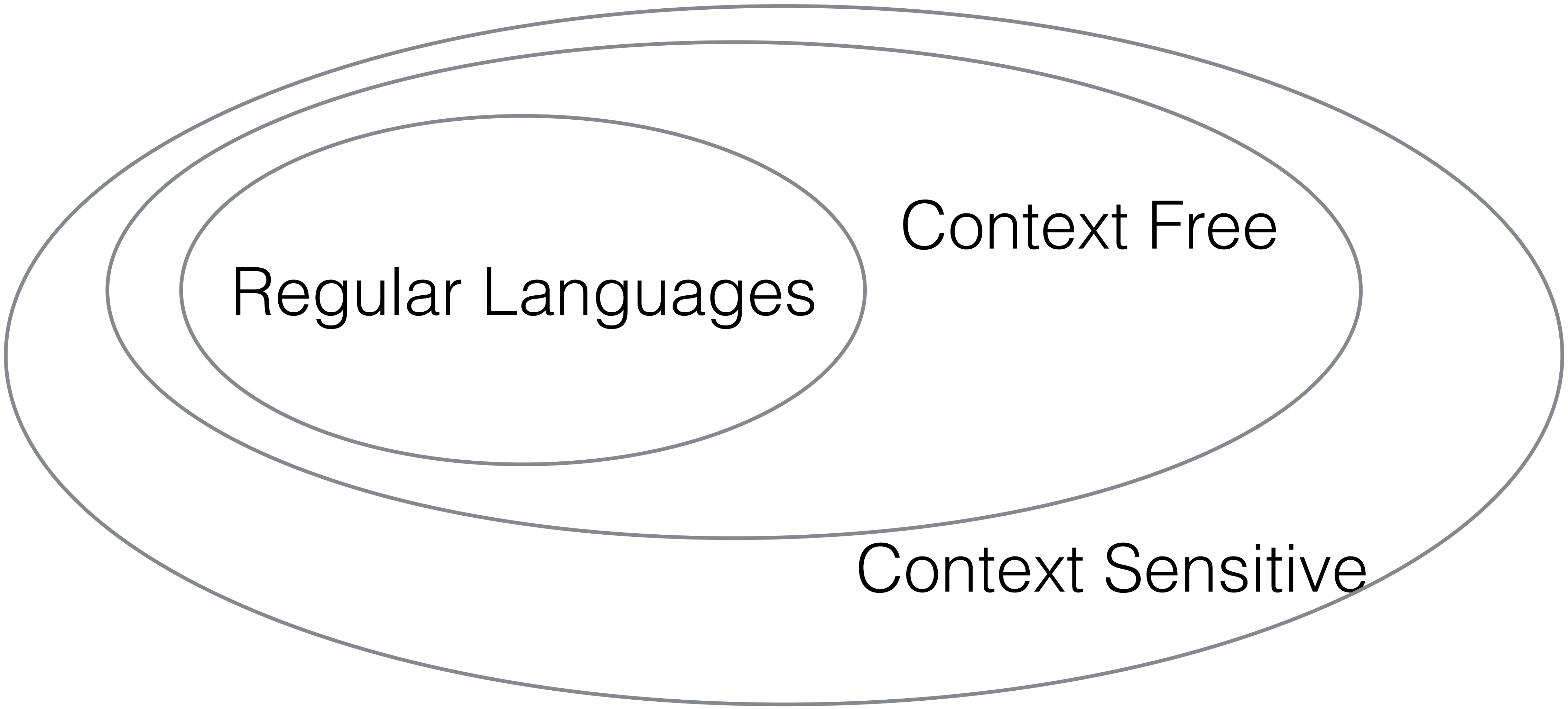
Chomsky Hierarchy



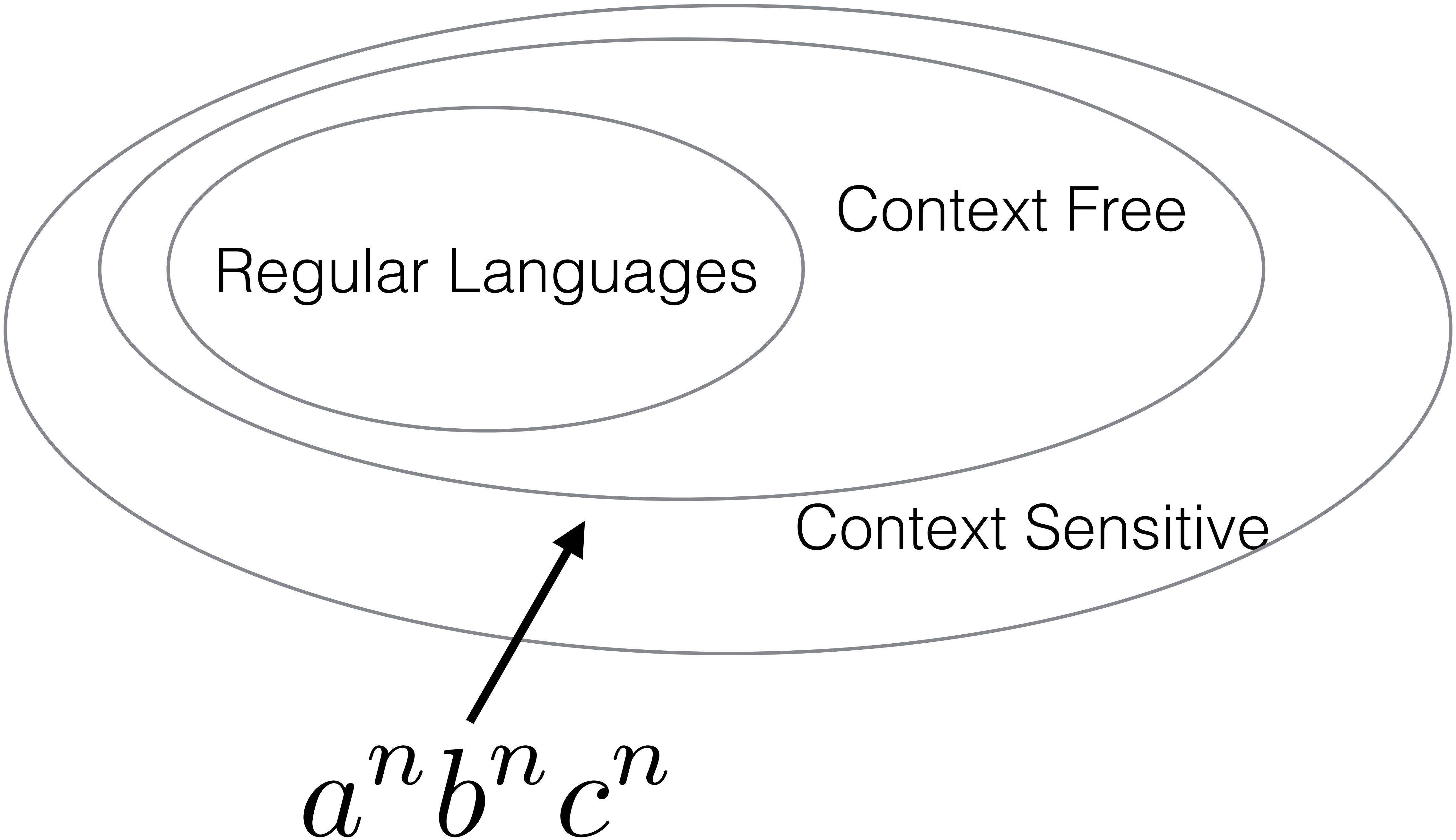
Chomsky Hierarchy



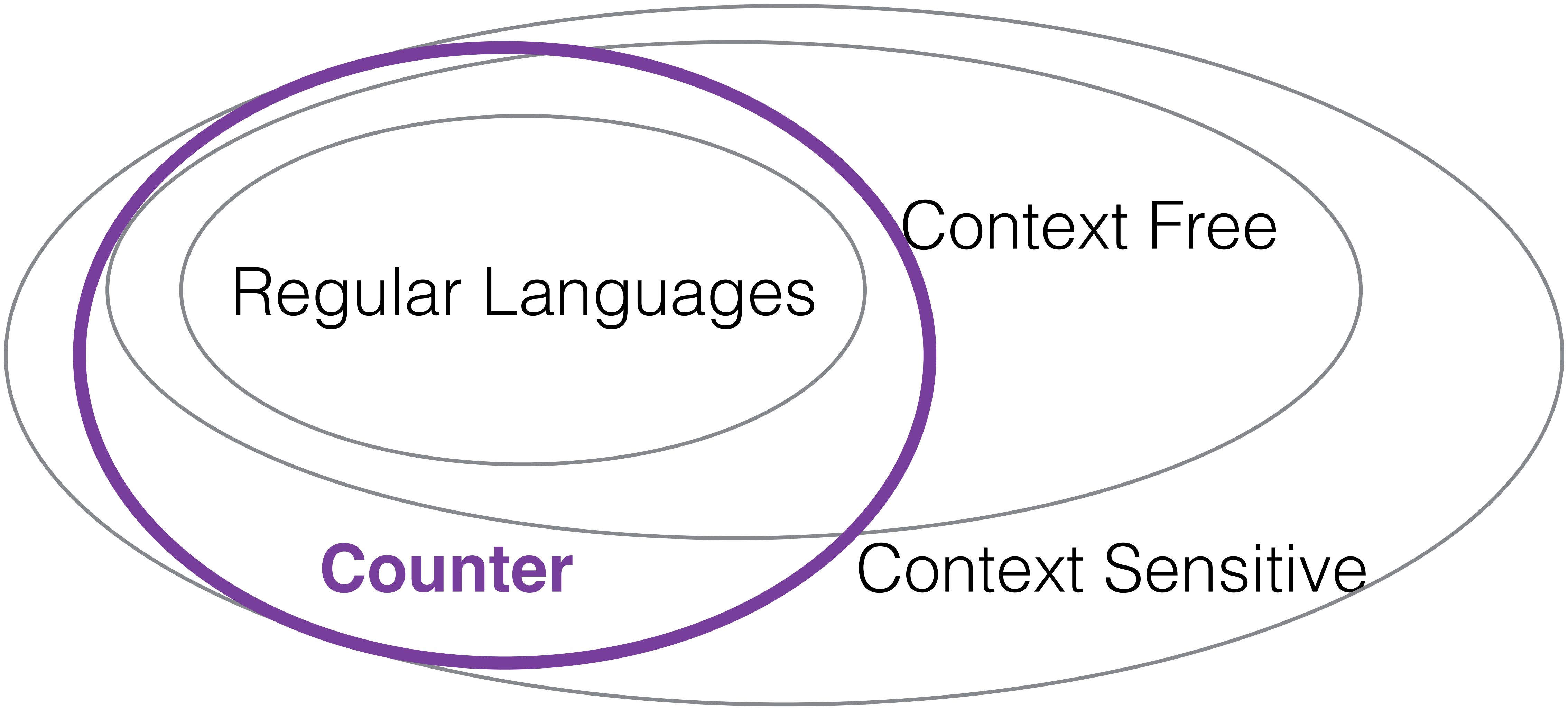
Chomsky Hierarchy



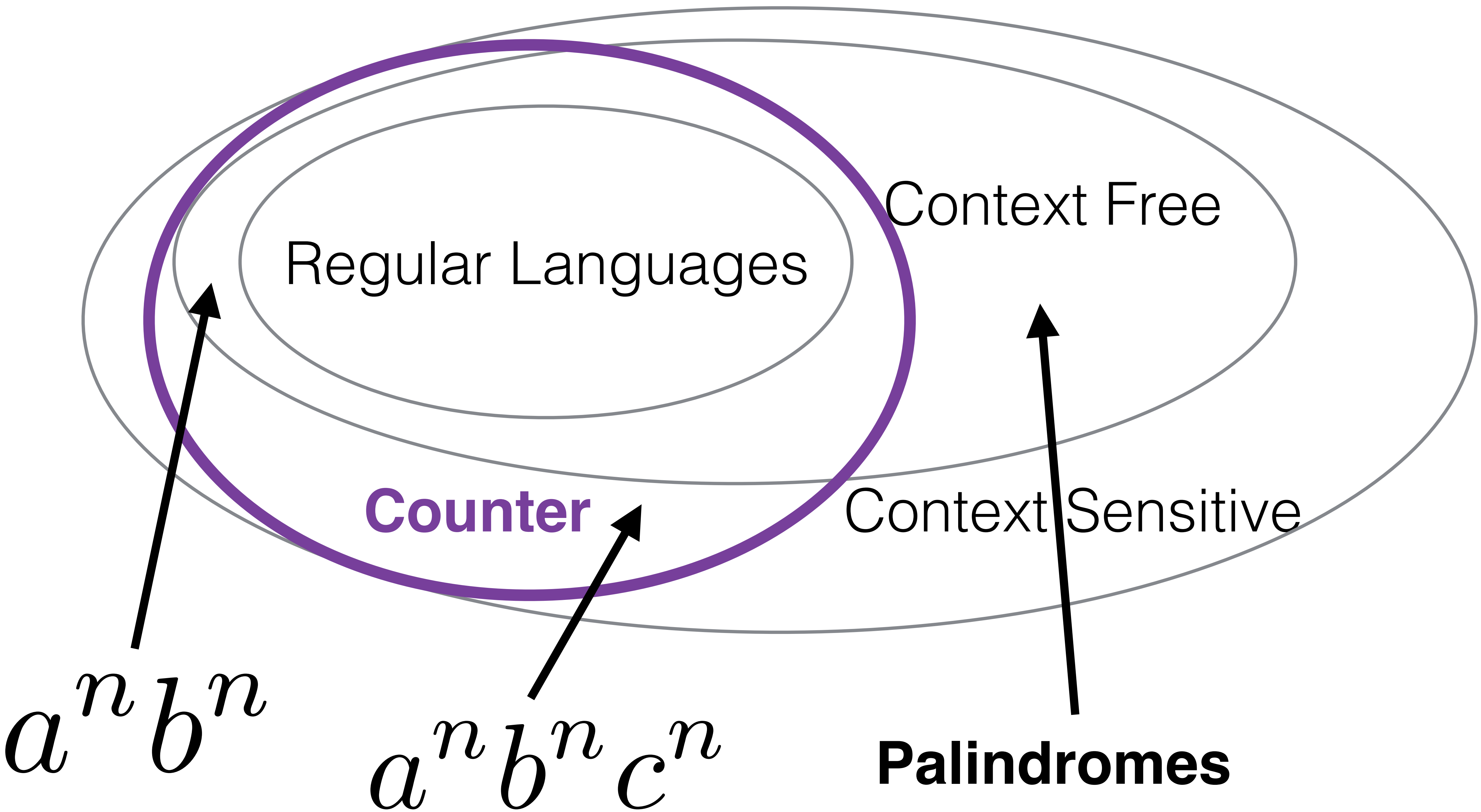
Chomsky Hierarchy



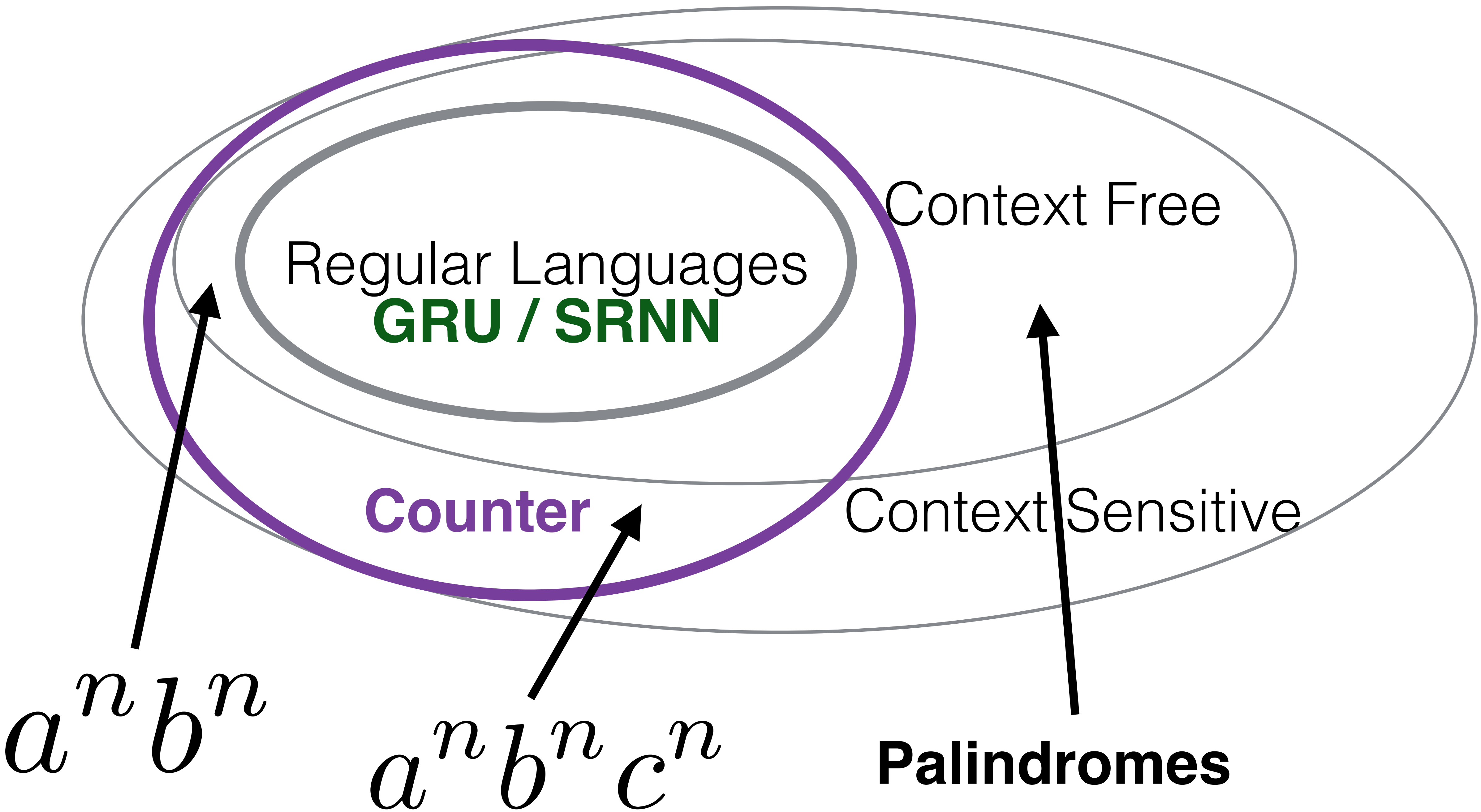
Power of Counting



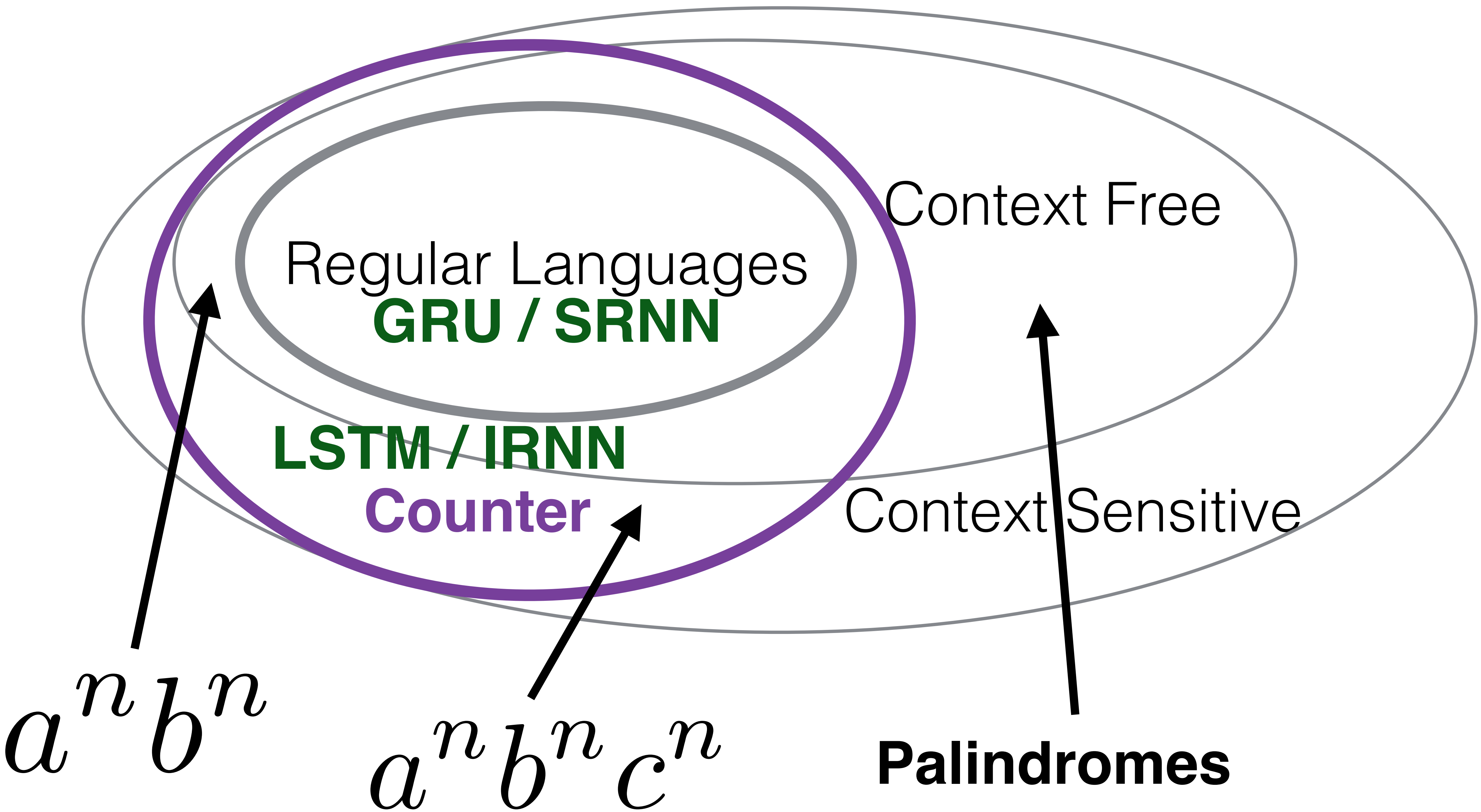
Power of Counting



Power of Counting



Power of Counting



IRNN / LSTM can count

$$\begin{aligned}
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\
\tilde{c}_t &= \tanh(W^c x_t + U^c h_{t-1} + b^c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ g(c_t)
\end{aligned}$$

IRNN / LSTM can count

$$
\begin{aligned}
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\
\tilde{c}_t &= \tanh(W^c x_t + U^c h_{t-1} + b^c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ g(c_t)
\end{aligned}
$$

1
(via sigmoid)

-1 , 1
(via tanh)

compare to zero is easy

IRNN / LSTM can count

counting is **EASY**!
just needs to saturate 3 gates.

$$
\begin{aligned}
f_t &= \sigma(W^f x_t + U^f h_{t-1} + b^f) \\
i_t &= \sigma(W^i x_t + U^i h_{t-1} + b^i) \\
o_t &= \sigma(W^o x_t + U^o h_{t-1} + b^o) \\
\tilde{c}_t &= \tanh(W^c x_t + U^c h_{t-1} + b^c) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ g(c_t)
\end{aligned}
$$

1
(via sigmoid)

-1 , 1
(via tanh)

compare to zero is easy

IRNN / LSTM can count

IRNN

$$h_t = max(0, (Wx_t + Uh_{t-1} + b))$$

+1 in one dim =INC
+1 in other dim =DEC

compare to zero
by subtracting dims
(requires MLP)

SRNN / GRU cannot count

SRNN

$$h_t = \tanh(Wx_t + Uh_{t-1} + b)$$

squashing prevents counting

SRNN / GRU cannot count

GRU

$$
\begin{aligned}
z_t &= \sigma(W^z x_t + U^z h_{t-1} + b^z) \\
r_t &= \sigma(W^r x_t + U^r h_{t-1} + b^r) \\
\tilde{h}_t &= \tanh(W^h x_t + U^h (r_t \circ h_{t-1}) + b^h) \\
h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t
\end{aligned}
$$

gate tie prevents counting

-1 , 1
(via tanh)

SRNN / GRU cannot count

can do some bounded counting within the -1,1 range.
hard: requiring precise setting of non-saturated values.

$$
\begin{aligned}
z_t &= \sigma(W^z x_t + U^z h_{t-1} + b^z) \\
r_t &= \sigma(W^r x_t + U^r h_{t-1} + b^r) \\
\tilde{h}_t &= \tanh(W^h x_t + U^h (r_t \circ h_{t-1}) + b^h) \\
h_t &= z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t
\end{aligned}
$$

gate tie prevents counting

-1 , 1
(via tanh)

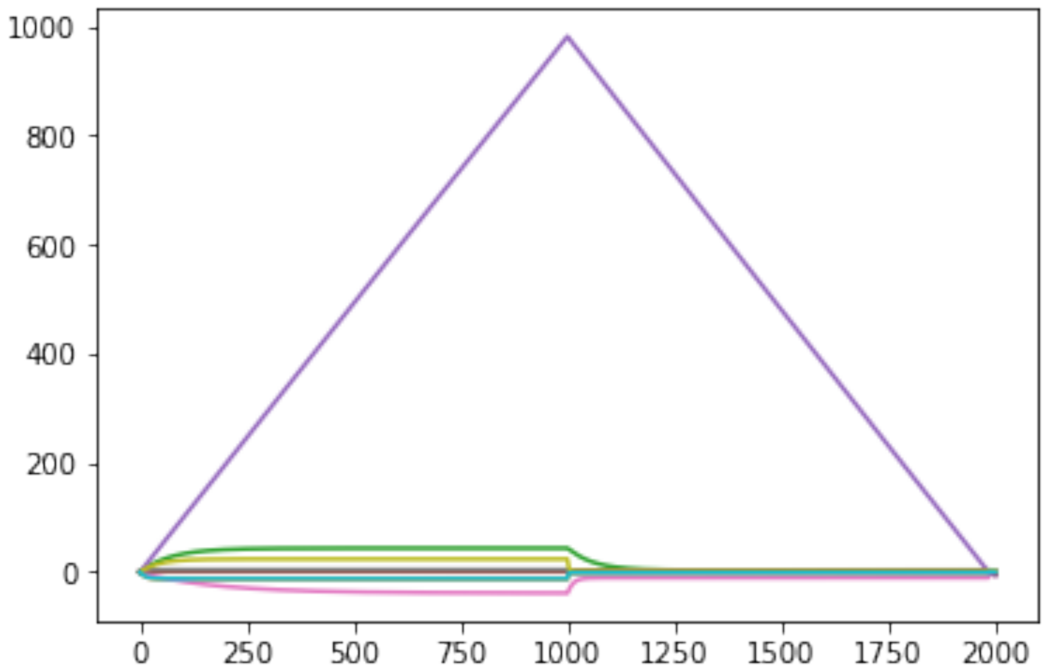
Counting in some other way?

cannot implement a binary-counter (or any k-base counter) in a single SRNN step.

LSTM vs. GRU

train on $\mathbf{a^n b^n}$ up to n=100

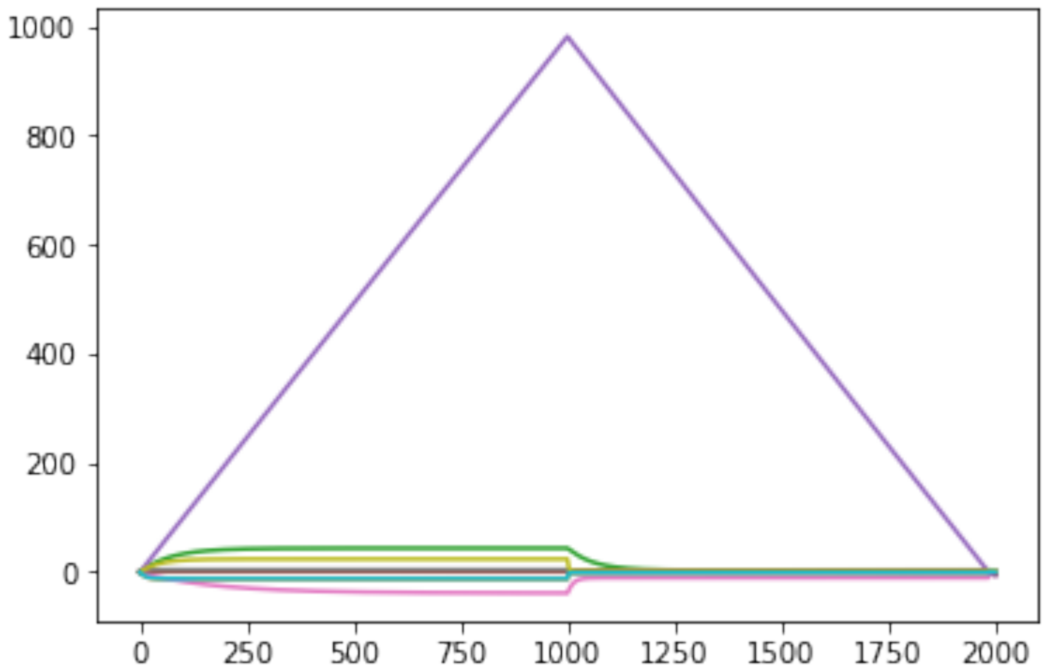
LSTM vs. GRU



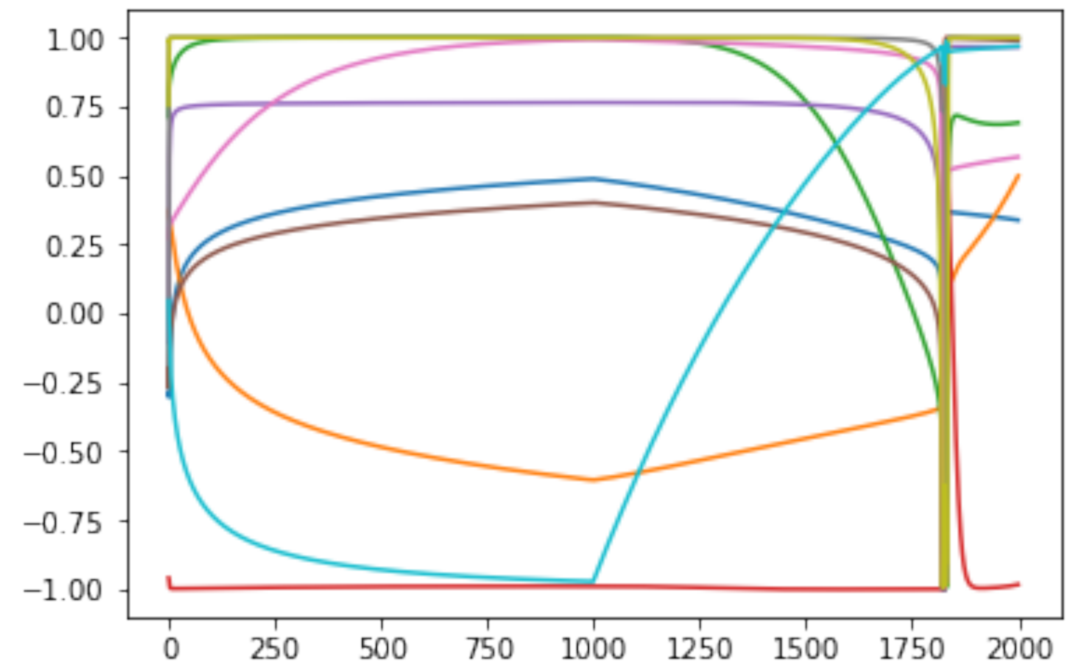
(a) $a^n b^n$-LSTM on $a^{1000} b^{1000}$

train on **$a^n b^n$** up to n=100

LSTM vs. GRU



(a) $a^n b^n$-LSTM on $a^{1000}b^{1000}$

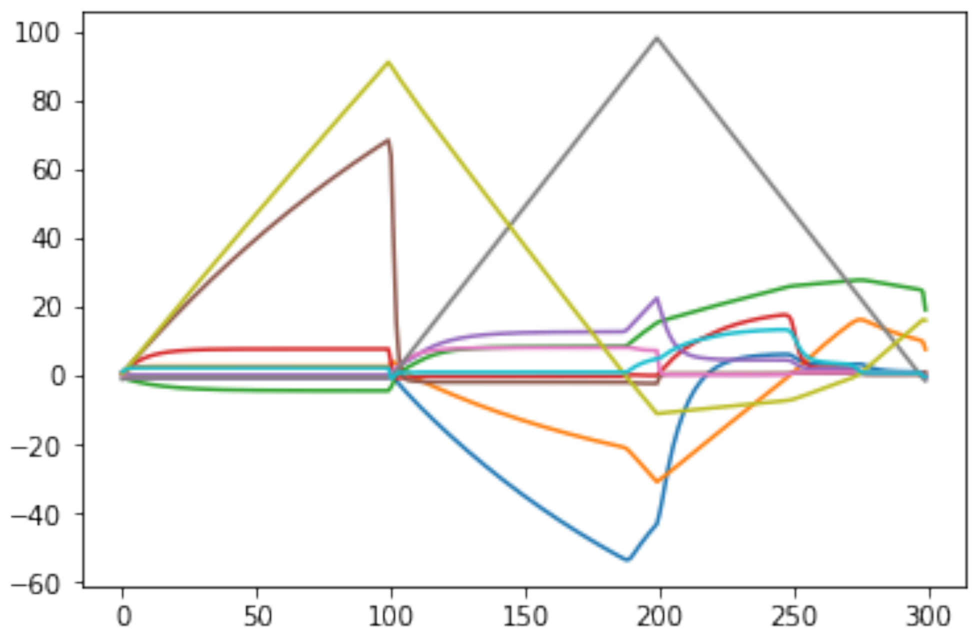


(c) $a^n b^n$-GRU on $a^{1000}b^{1000}$

train on **$a^n b^n$** up to n=100

GRU starts to fail at n=38

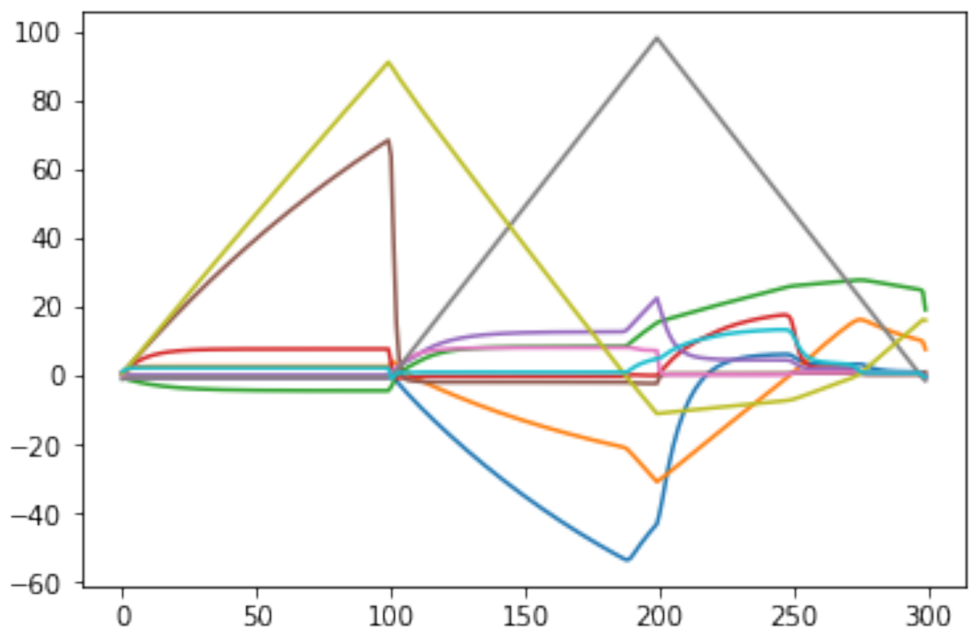
LSTM vs. GRU



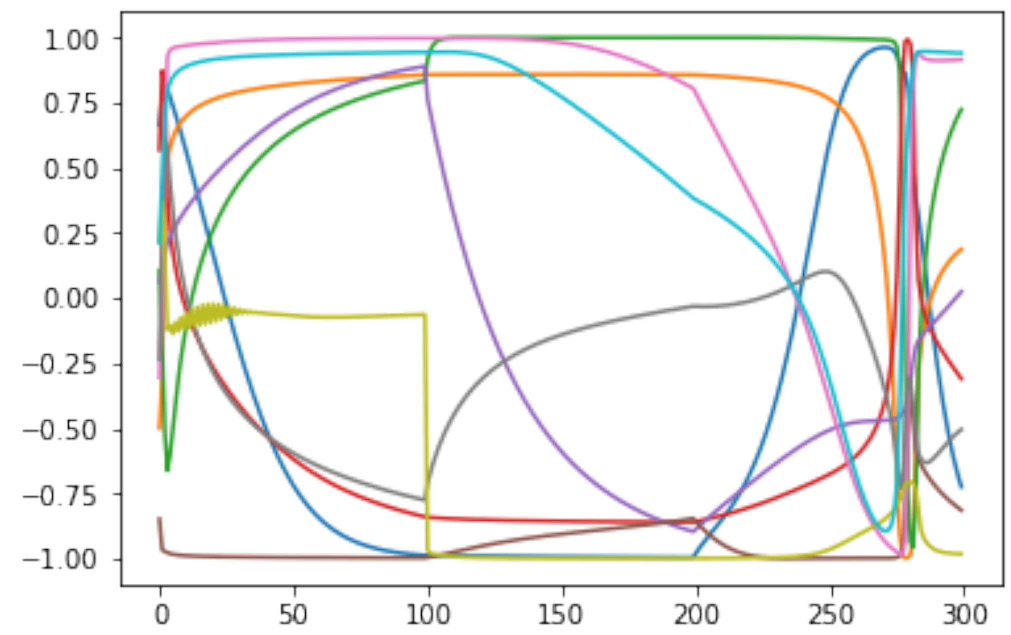
(b) $a^n b^n c^n$-LSTM on $a^{100}b^{100}c^{100}$

train on $\mathbf{a^n b^n c^n}$ up to n=50

LSTM vs. GRU



(b) $a^n b^n c^n$-LSTM on $a^{100} b^{100} c^{100}$



(d) $a^n b^n c^n$-GRU on $a^{100} b^{100} c^{100}$

train on **$a^n b^n c^n$** up to n=50

GRU starts to fail at n=8

To summarize (this part)

- Escape Turing-completeness by looking into finite-precision, real-time RNN
- Real difference in expressive power between [SRNN, GRU] and [IRNN, LSTM].
- Small architectural choices can matter.

Q6: Extracting a discrete representation from a trained model.

what do trained LSTM acceptors encode?

Extracting FSAs from RNNs

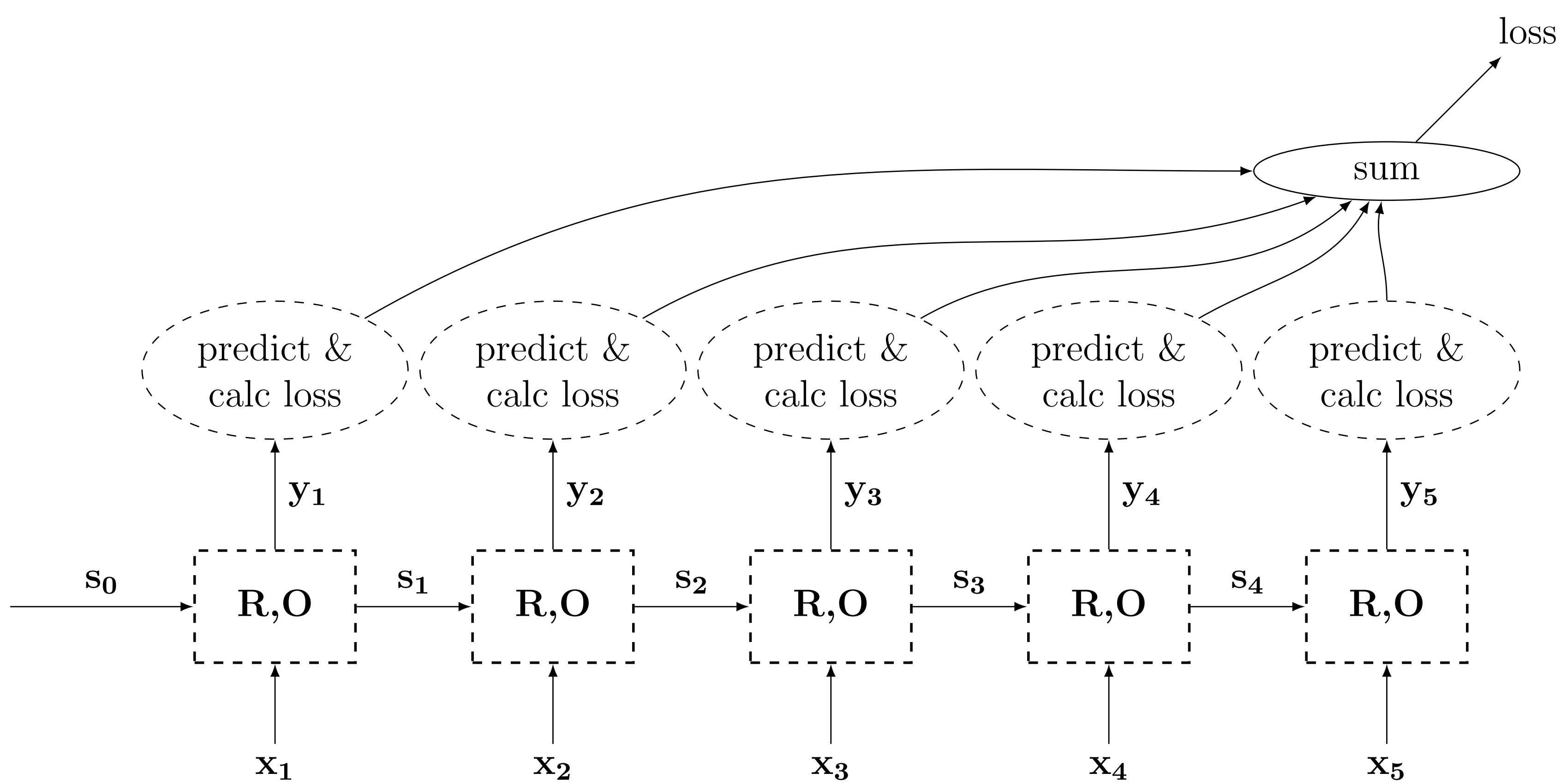
Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples

Gail Weiss[1], Yoav Goldberg[2], and Eran Yahav[1]

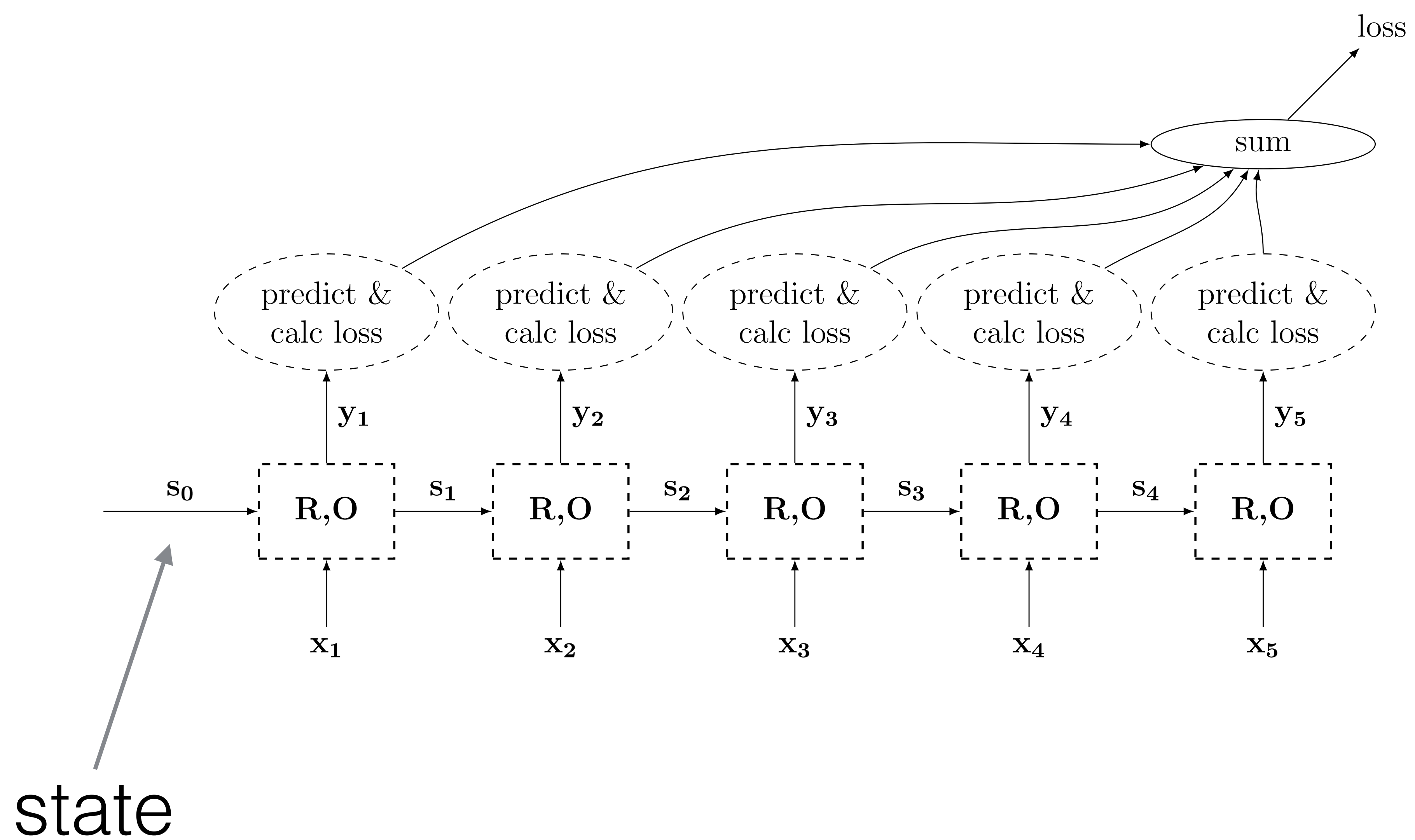
(ICML 2018)

RNN acceptors as State Machines

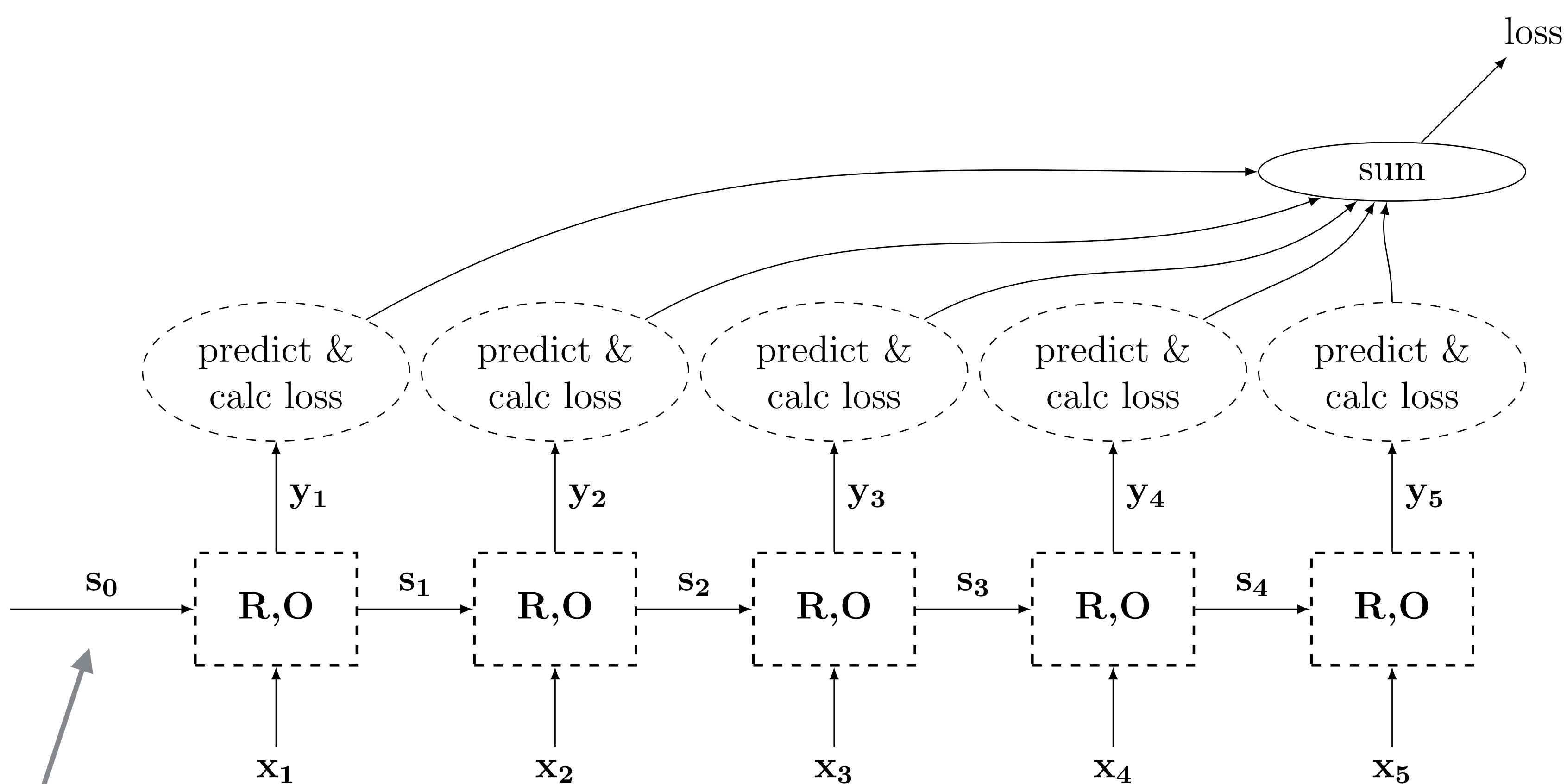
RNN acceptors as State Machines



RNN acceptors as State Machines



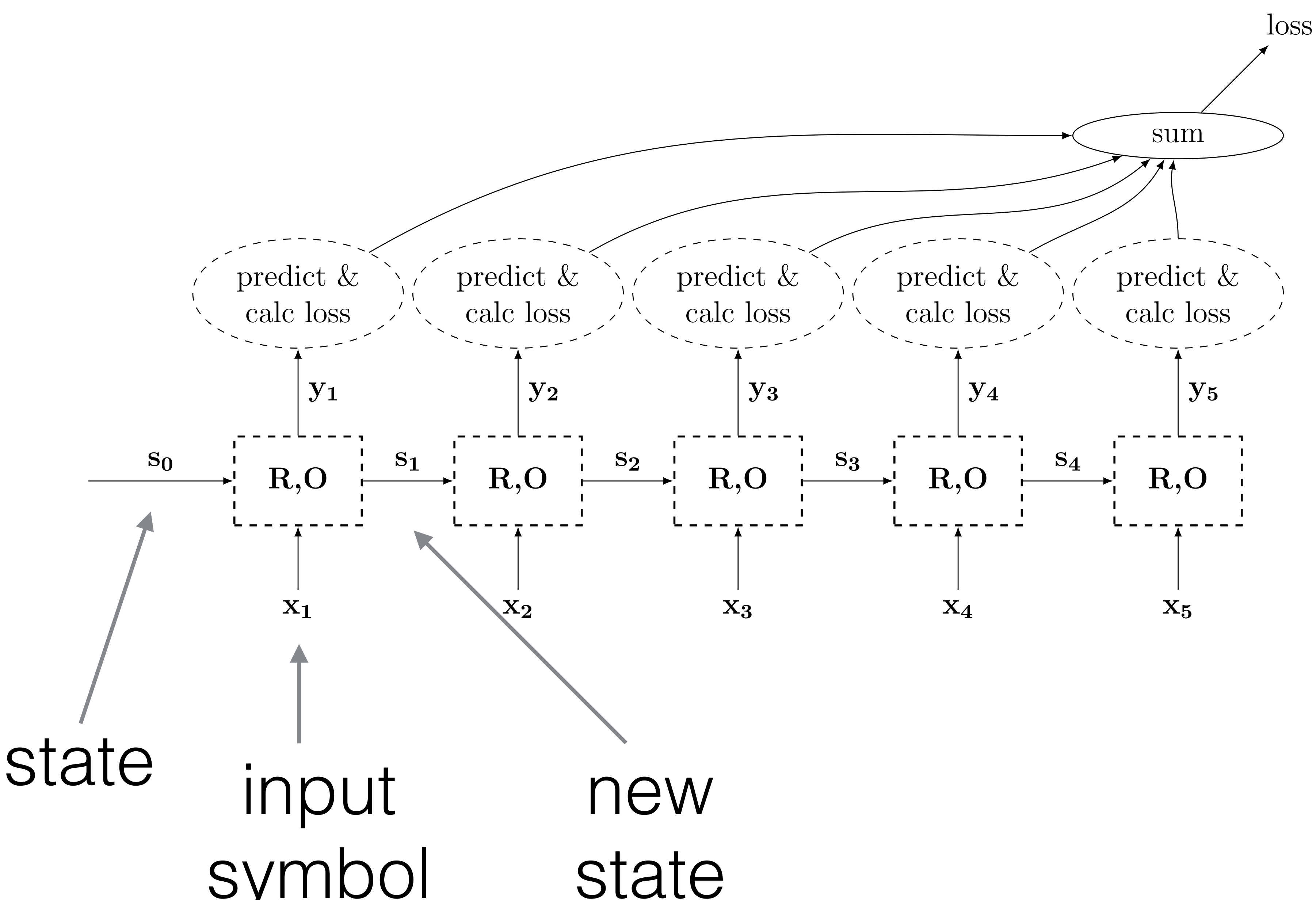
RNN acceptors as State Machines



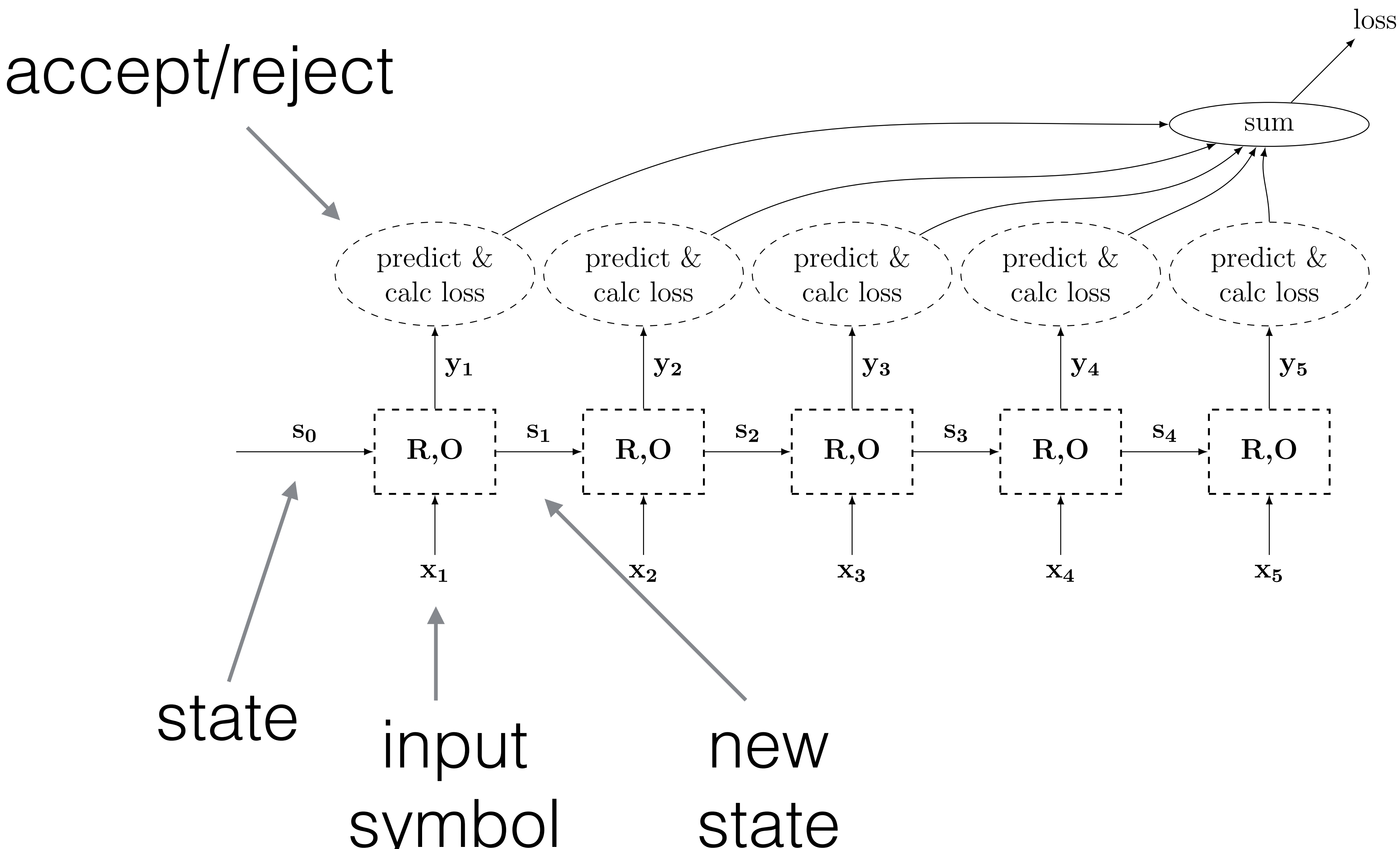
state

input symbol

RNN acceptors as State Machines

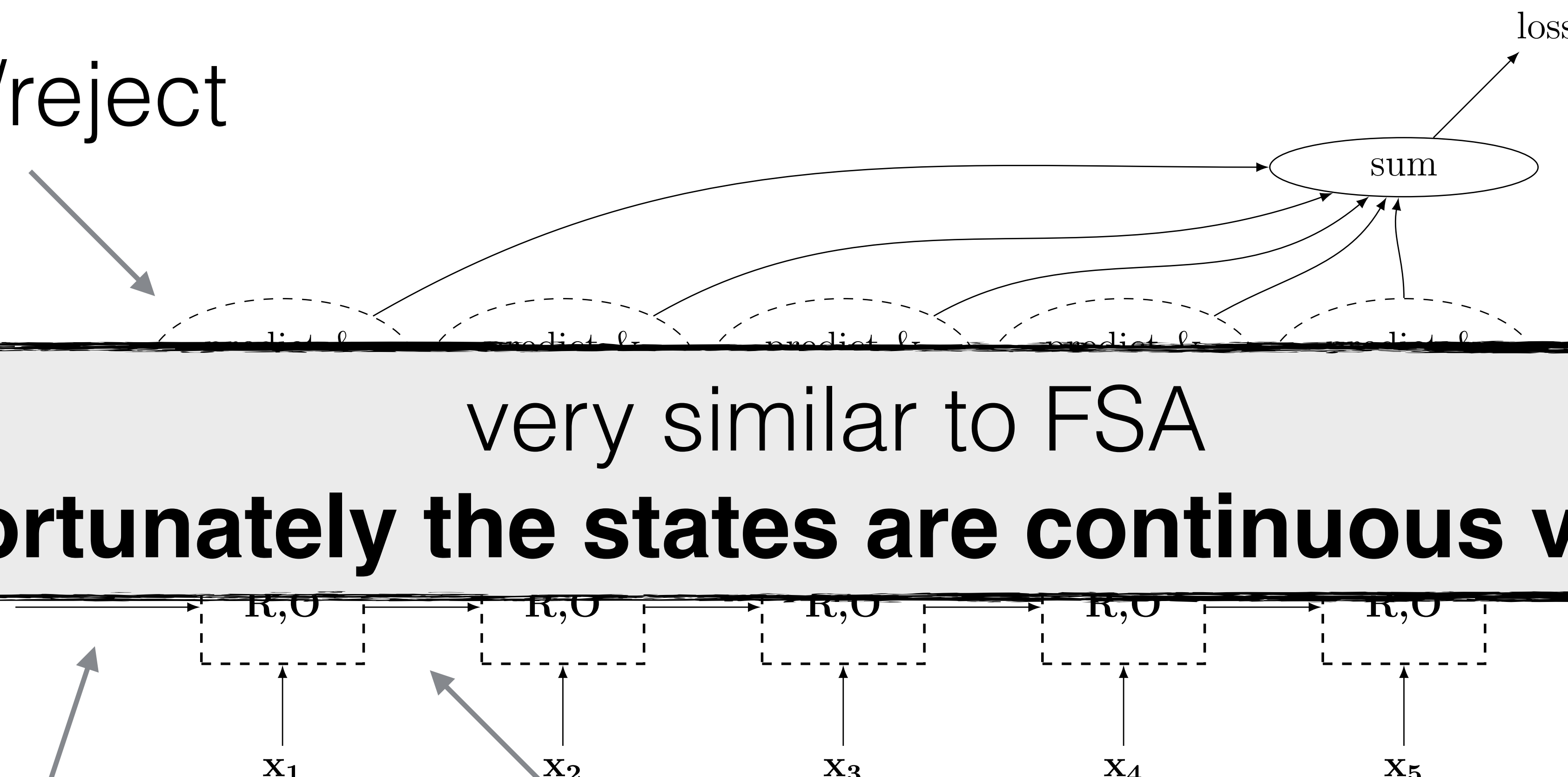


RNN acceptors as State Machines



RNN acceptors as State Machines

accept/reject



very similar to FSA

unfortunately the states are continuous vectors

state

input symbol

new state

INFORMATION AND COMPUTATION 75, 87–106 (1987)

Learning Regular Sets from Queries and Counterexamples*

DANA ANGLUIN

Department of Computer Science, Yale University, P.O. Box 2158, Yale Station, New Haven, Connecticut 06520

Learning Finite State Automata

- **L* algorithm**
  - FSAs are learnable from "**minimally adequate teacher**"
    - **Membership queries**

      "does this word belong in the language?**"**
    - **Equivalence queries**

      "does this automaton represent the language?"

Game Plan

- Train an RNN
- Use it as a Teacher in the L* algorithm
- L* learns the FSA represented by the RNN

RNN as Minimally Adequate Teacher

Membership Queries

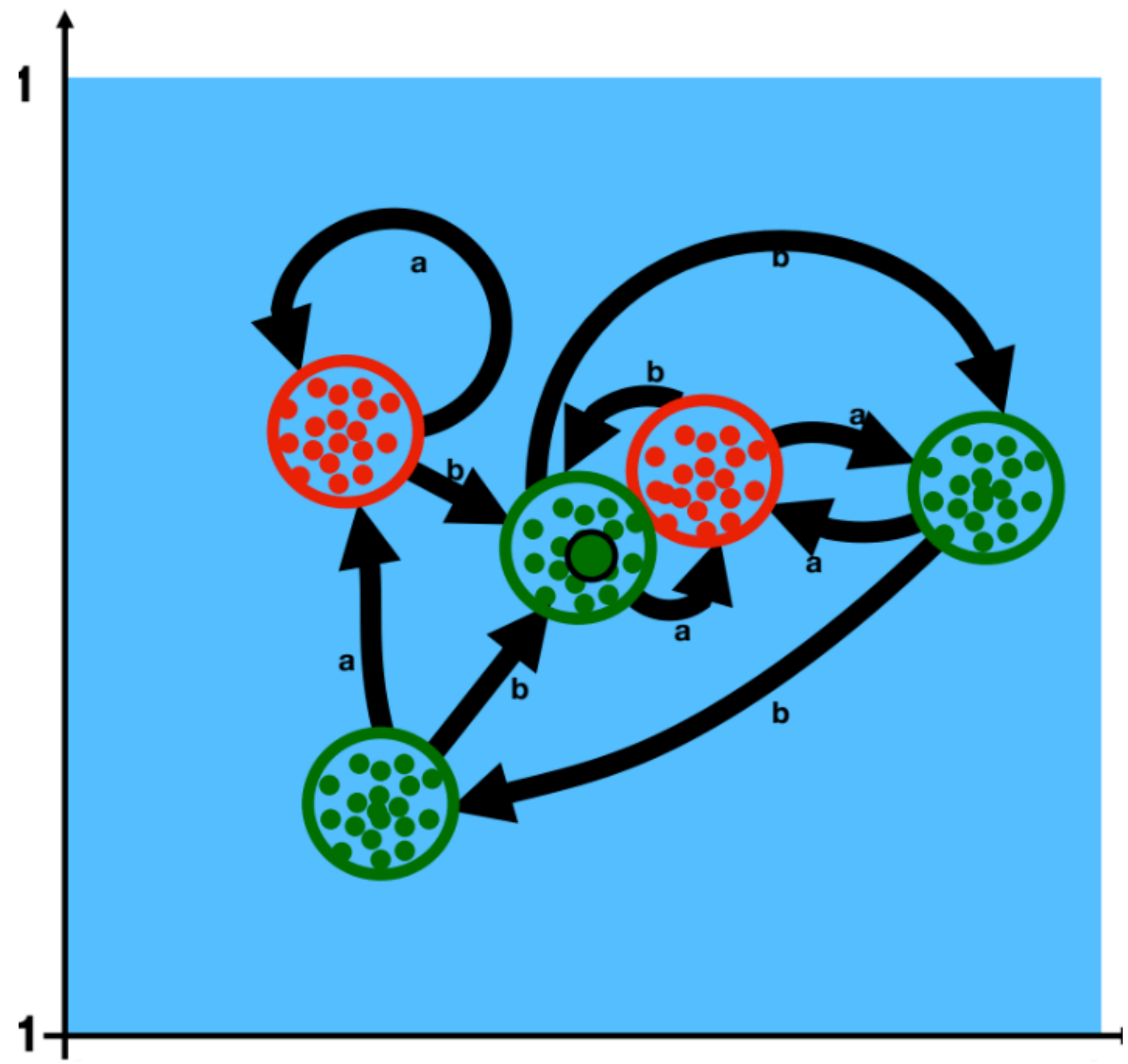
Easy. Just run the word through the RNN.

Equivalence Queries

Hard. Requires some trickery.

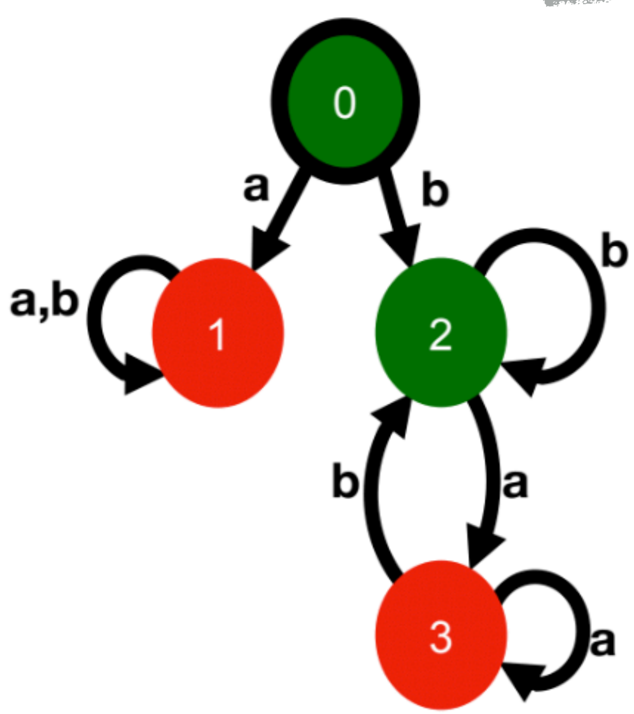
Answering Equivalence Queries

- Map RNN states to discrete states, forming an FSA abstraction of the RNN.

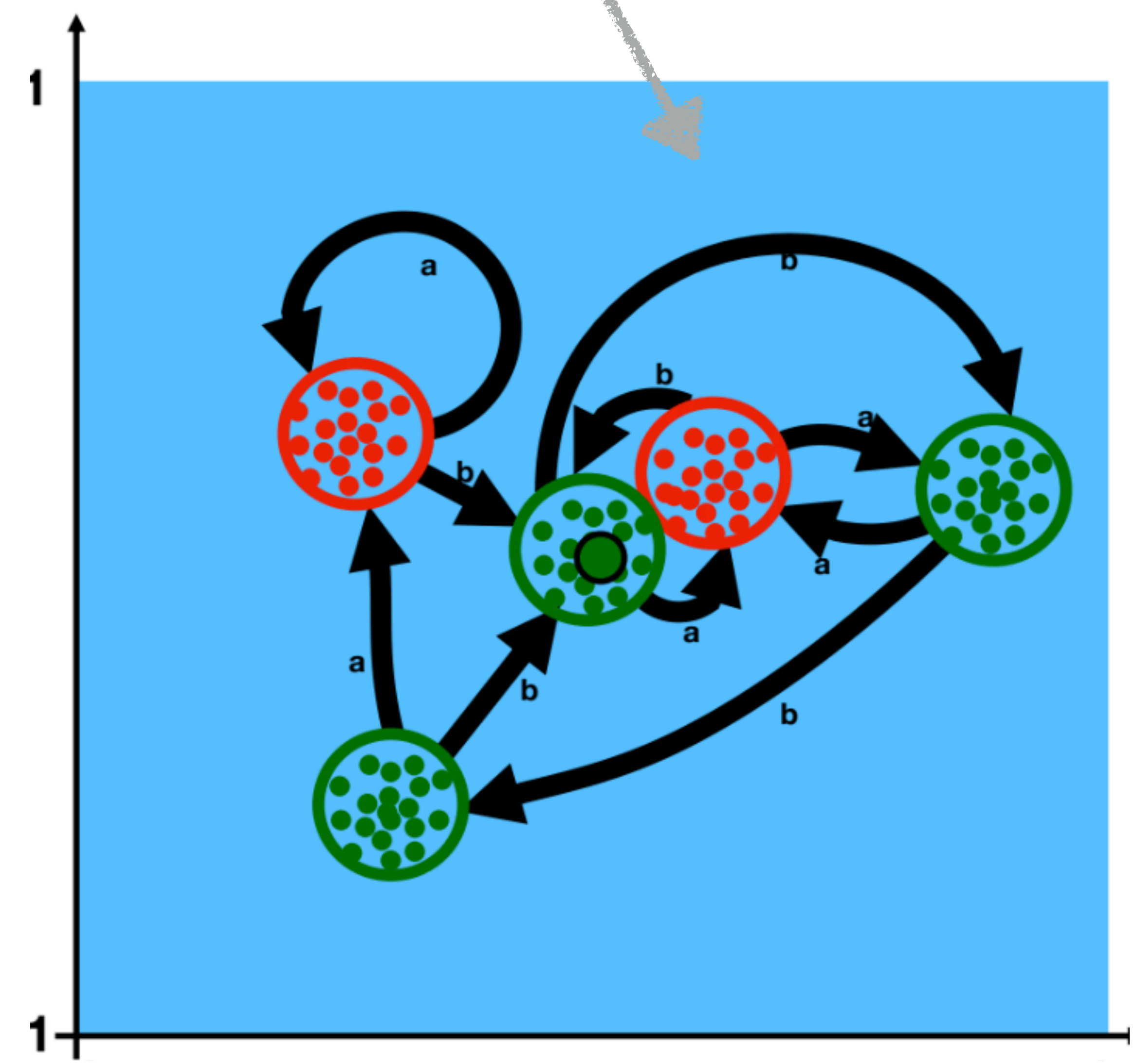


Answering Equivalence Queries

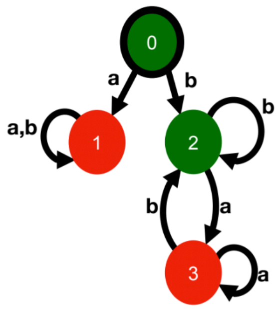
- Compare L* **Query FSA** to **RNN-Abstract-FSA**.

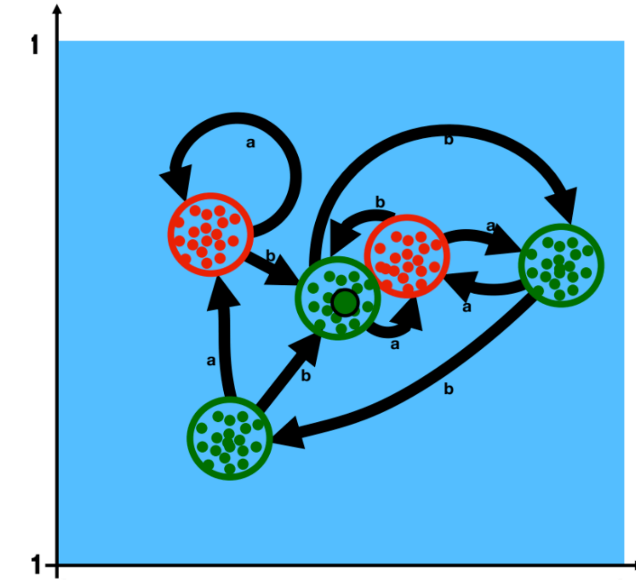


=??



Answering Equivalence Queries





- **Conflict**?
  - Maybe state-mapping is wrong.
    If so: **refine the mapping**.
  - Maybe L* FSA is wrong.
    If so: **return a counter example**.

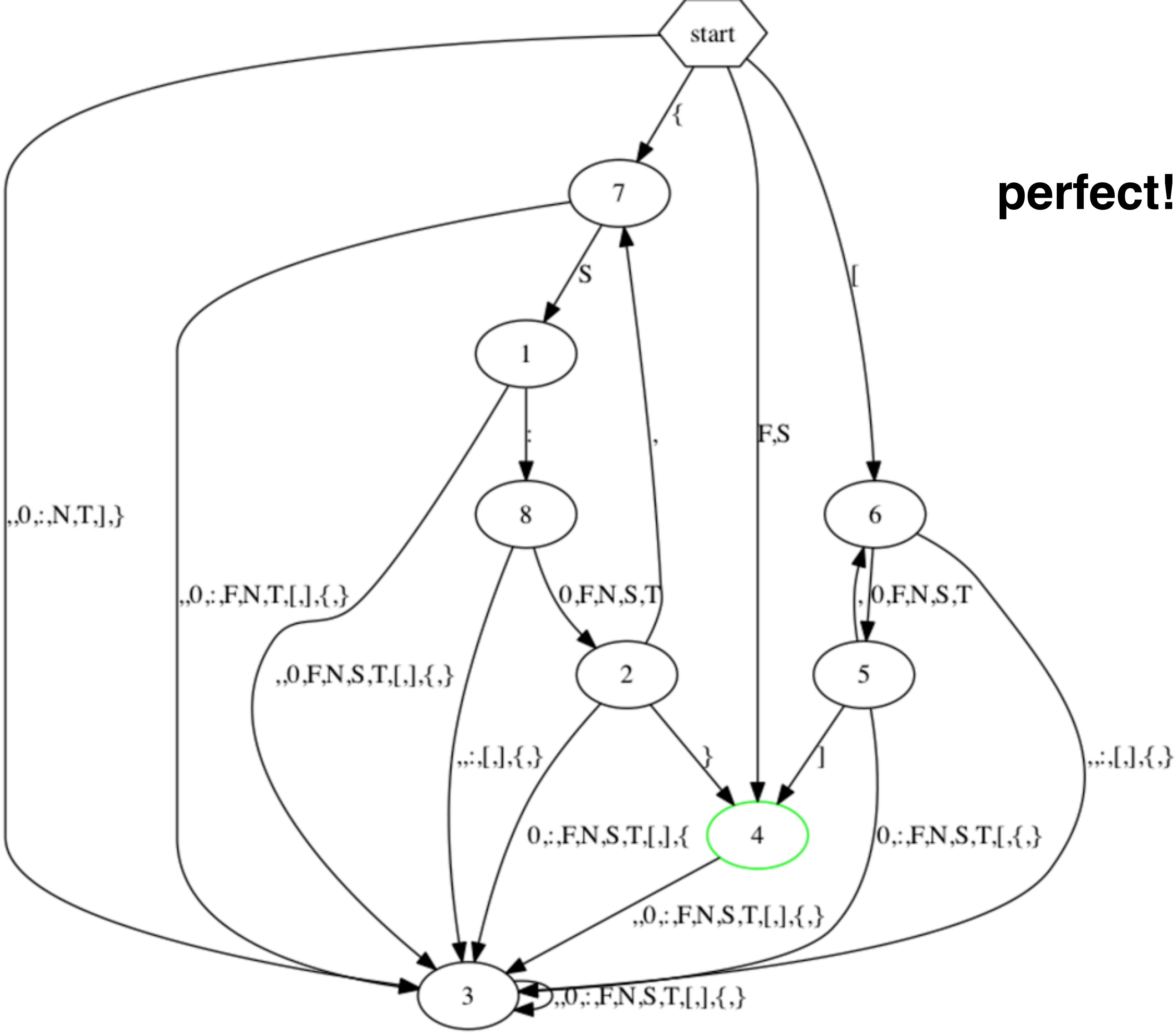
Some Results

- **Many random FSAs:**
    - 5 or 10 states, alphabet sizes of 3 or 5
- LSTM/GRU with 50, 100, 500 dimensions.
- The FSAs were **learned well** by LSTM / GRU
- And **recovered well** by L*.

"lists or dicts"

- `F`
- `S`
- `[F,S,0,F,N,T]`
- `{S:F,S:F,S:0,S:T,S:S,S:N}`

alphabet: `F S 0 N T , : { } [ ]`


start
{
7
S
1
:
8
,
F,S
[
6
,,0,:,N,T,],}
,,0,:,F,N,T,[,],{,}
0,F,N,S,T
,
0,F,N,S,T
,,0,F,N,S,T,[,],{,}
2
5
,,:,[,],{,}
}
]
,,:,[,],{,}
0,:,F,N,S,T,[,],{
4
0,:,F,N,S,T,[,{,}
,,0,:,F,N,S,T,[,],{,}
3
,,0,:,F,N,S,T,[,],{,}

perfect!

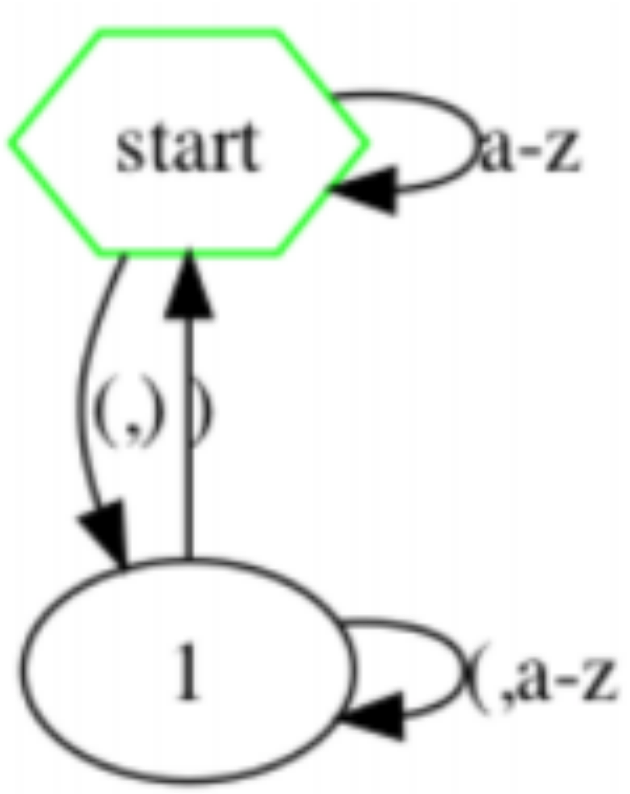
Balanced Parenthesis

`(a((ejka((acs))(asdsa))djljf)kls(fjkljklkids))`

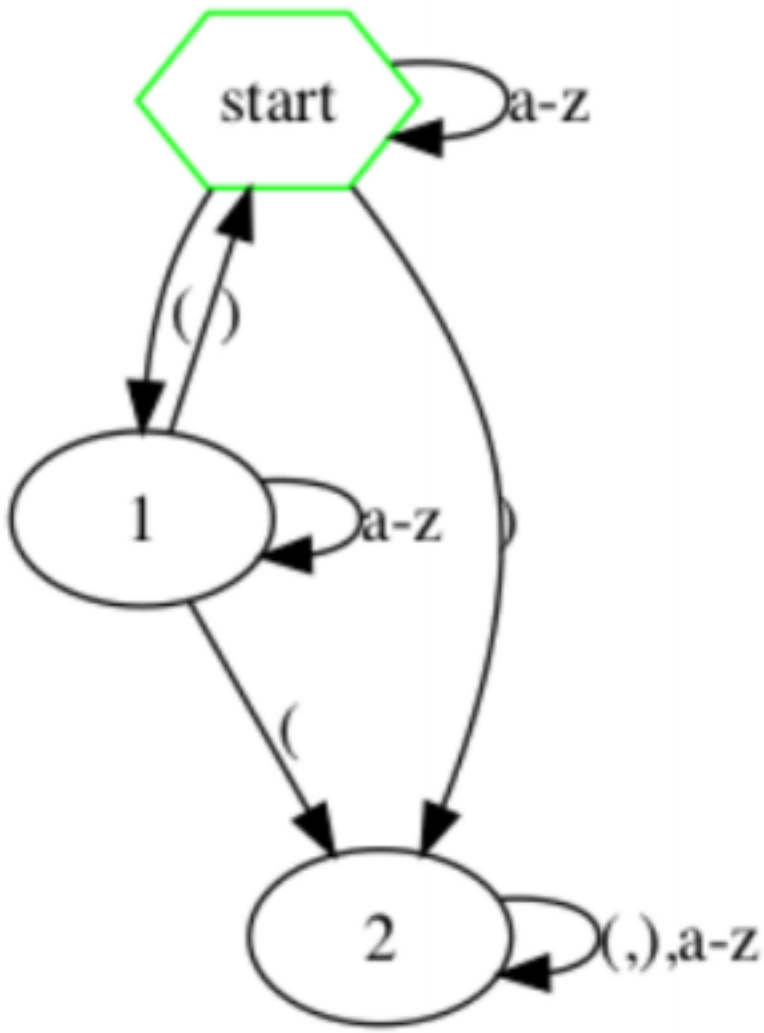
alphabet: `a-z ( )`

nesting level up to 8.

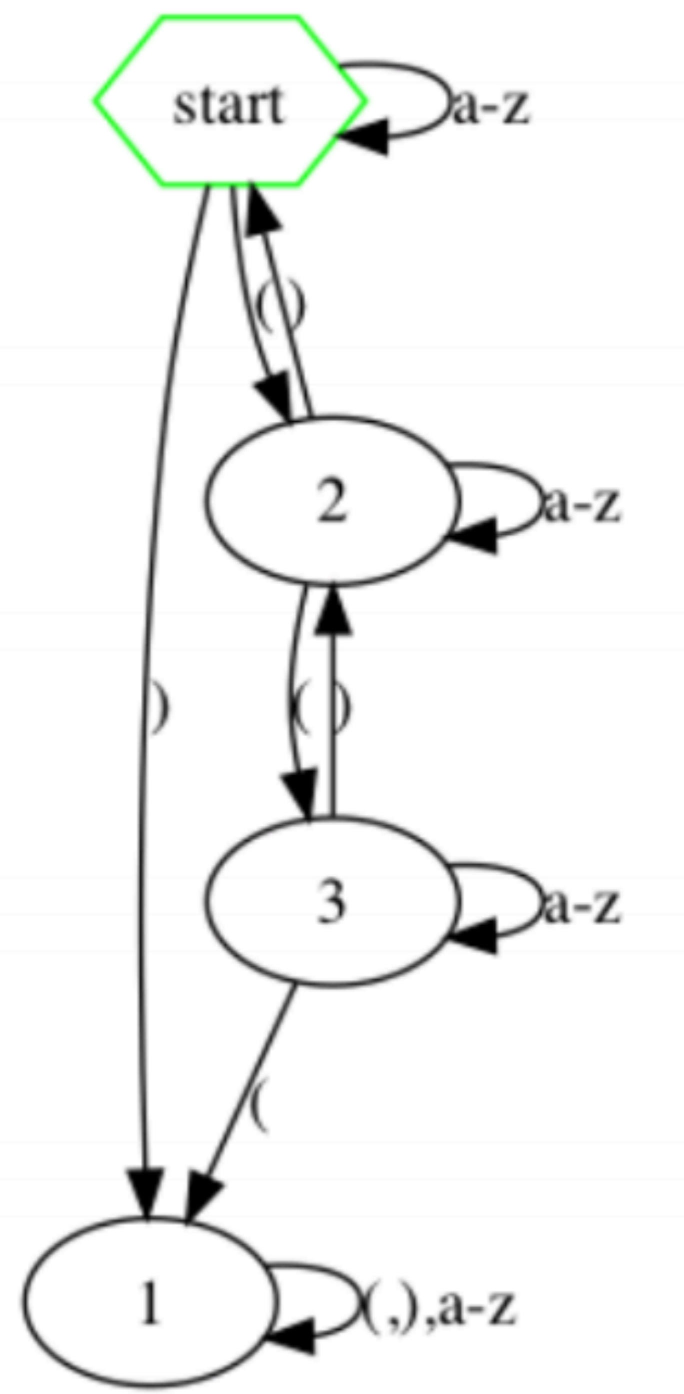
Balanced Parenthesis



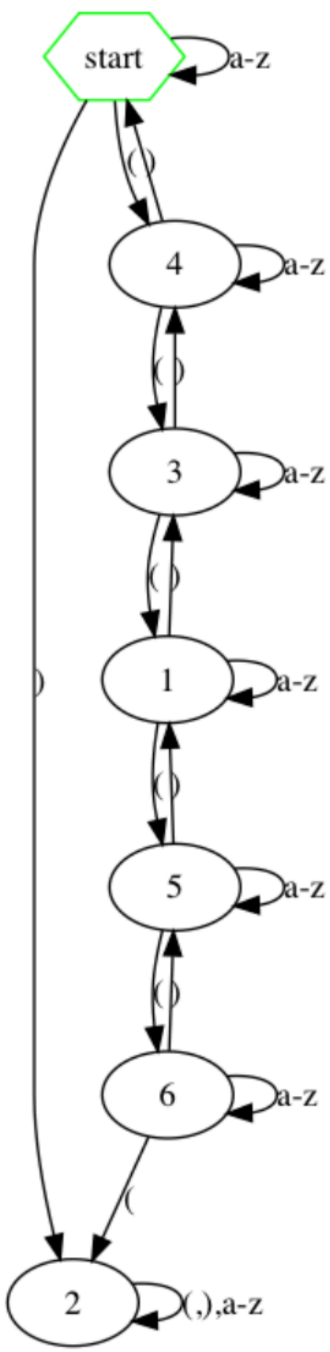
Balanced Parenthesis



Balanced Parenthesis

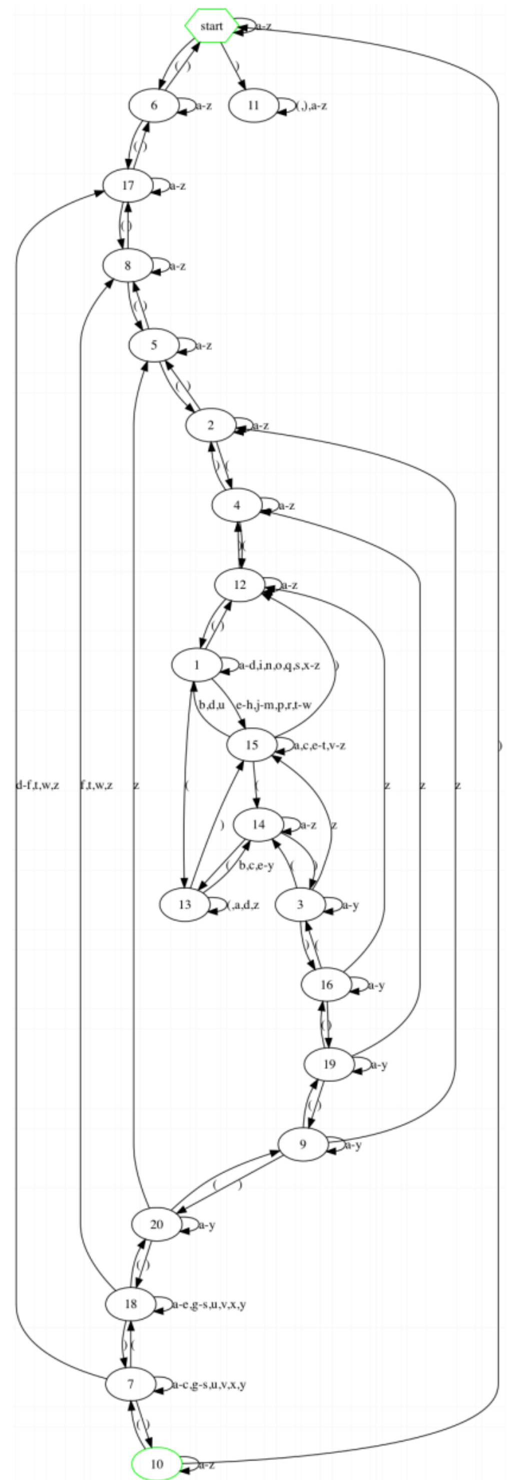


Balanced Parenthesis



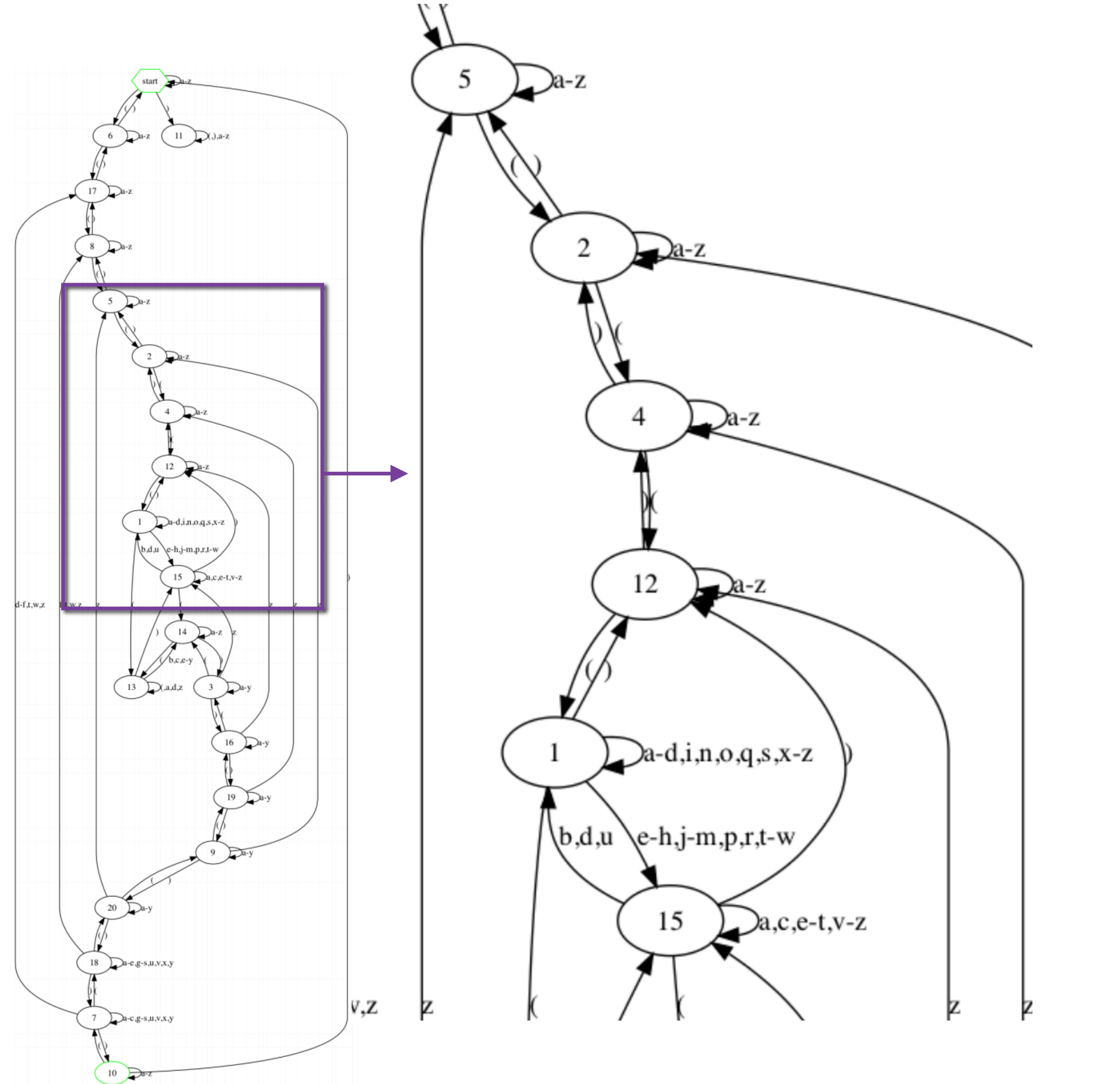
Balanced Parenthesis

final automaton:



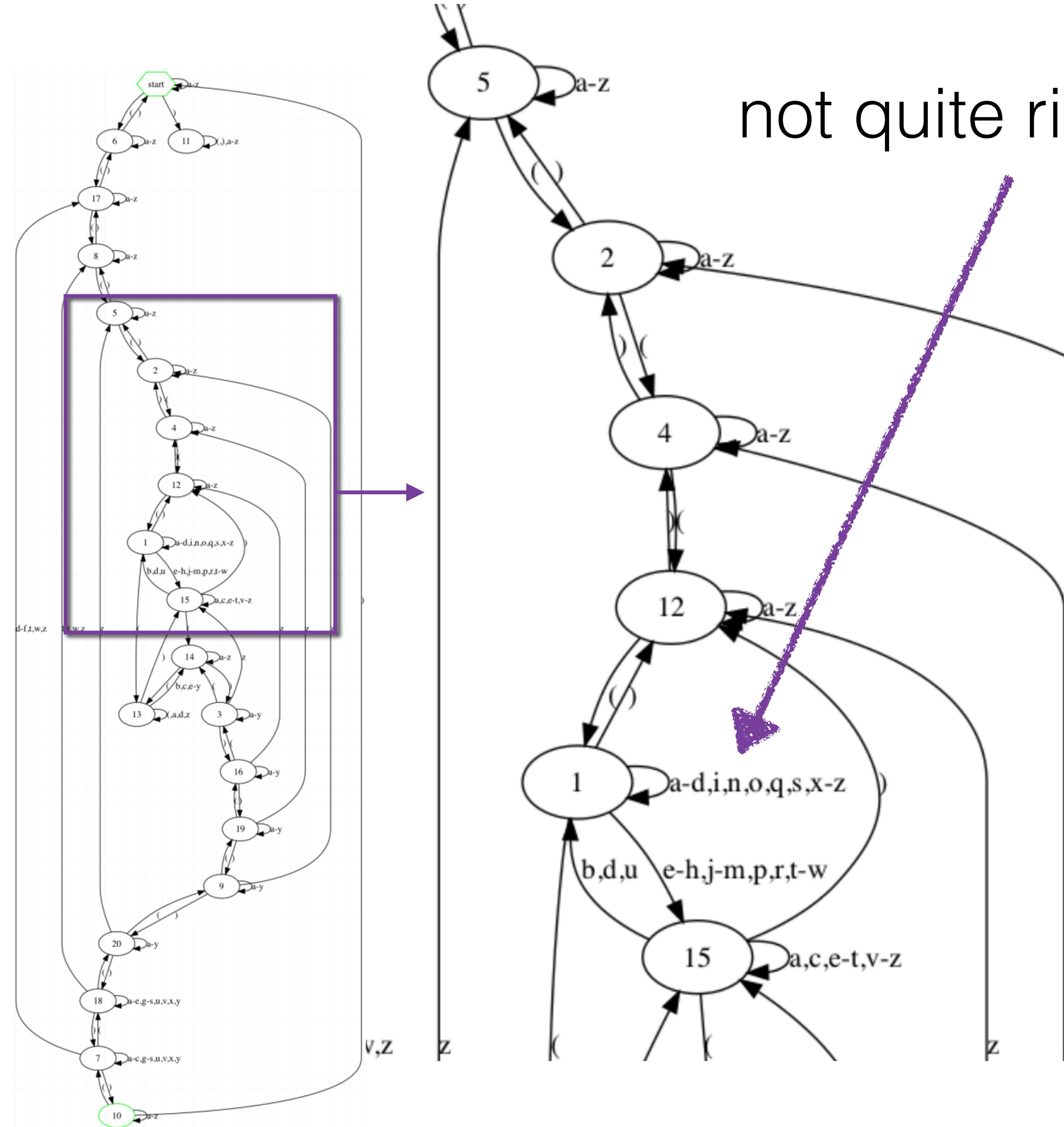
Balanced Parenthesis

final automaton:



Balanced Parenthesis

final automaton:



not quite right

"Emails"

- bla12@abc.com, ahjlkoo@jjjgs.net

```
[a-z][a-z0-9]*@[a-z0-9]+\.(com|net|co\.[a-z][a-z])
```

"Emails"

- bla12@abc.com, ahjlkoo@jjjgs.net

[a-z][a-z0-9]*@[a-z0-9]+\.(com|net|co\.[a-z][a-z])

20,000 positive examples
20,000 negative examples
2,000 examples dev set

"Emails"

- `bla12@abc.com, ahjlkoo@jjjgs.net`

`[a-z][a-z0-9]*@[a-z0-9]+\.(com|net|co\.[a-z][a-z])`

20,000 positive examples
20,000 negative examples
2,000 examples dev set

LSTM has 100% accuracy on both train and dev (and test)

"Emails"

the extraction algorithm did not converge.
we stopped it when it reached over 500 states.

LSTM has 100% accuracy on both train and dev (and test)

"Emails"

the extraction algorithm did not converge.
we stopped it when it reached over 500 states.

some counter-examples it found:

25.net

5x.nem

2hs.net

LSTM has 100% accuracy on both train and dev (and test)

- **We can extract FSAs from RNNs**
  - ... if the RNN indeed captured a regular structure
  - ... and in many cases the representation captured by the RNN is much more complex (and wrong!) than the actual concept class.

- **Much more to do:**
  - scale to larger FSAs and alphabets
  - scale to non-regular languages
  - apply to "real" language data
  - ....

To summarize

To summarize

scratching the black box

To summarize

scratching the black box

To summarize

scratching the black box

- LSTMs (deep nets, Transformers, ...) are very powerful
  - We know how to use them.
  - We don't know enough about their power and limitations.
  - **Our intuitions are often wrong**.
  - We should try to understand them better.
  - Using **Algorithms**, using **Math**, or using **Science.**
  - **Very excited to see the evolving community around these questions. Join the fun.**

To summarize

scratching the black box

Thanks.
Questions?