Doing stuff with LSTMs

Yoav Goldberg

Dec 2017

CLIC-IT 2017

Deep Learning Revolution

IT LEARNS ON ITS OWN.

IT WORKS LIKE THE BRAIN.

IT CAN DO ANYTHING.

My experience with Deep Learning for Language

``I'm sorry Dave, I'm afraid I can't do that."

(not in the scary sense)

My experience with Deep Learning for Language

- With proper tools, easy to produce "innovative" models.
- Not so easy to get good results.
- With Feed-forward nets, hard to beat linear models w/ human engineered feature combinations.
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.

My experience with Deep Learning for Language

- With proper ... ative" models.
- Not so easy ...
- With Feed-f ... ar models w/ human engi ...
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.

May be different if you care to optimize parameters like crazy.
I don't have the resources nor the patience.

My experience with Deep Learning for Language

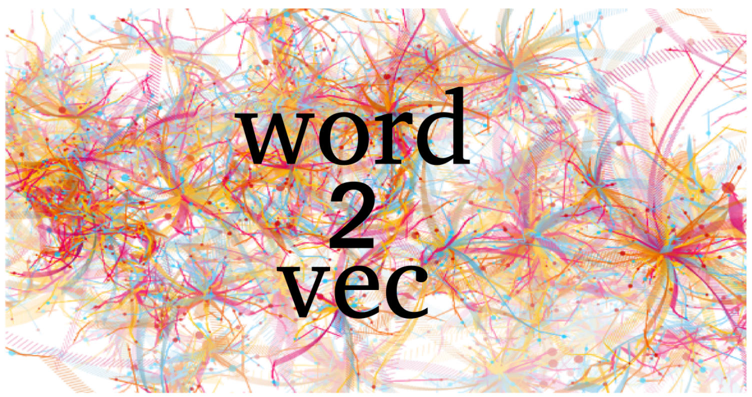
- With proper tools, easy to produce "innovative" models.
- Not so easy to get good results.
- With Feed-forward nets, hard to beat linear models w/ human engineered feature combinations.
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.
- Semi-sup learning sort-of easy with word-embeddings.

word2vec

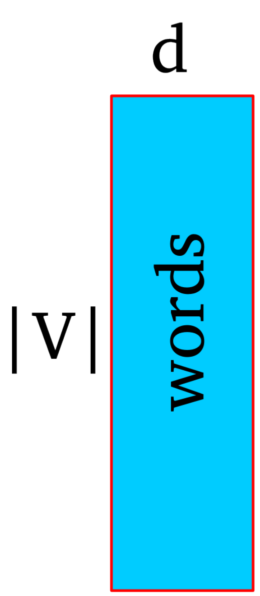
Text

Wikipedia

feed in text



wait a few hours



dog = (0.12,-0.32,0.92,0.43,-0.3 ...)
cat = (0.15,-0.29,0.90,0.39,-0.32 ...)

chair = (0.8,0.9,-0.76,0.29,0.52 ...)

get a |V|xd matrix W where each row is a vector for a word

- dog
  - cat, dogs, dachshund, rabbit, puppy, poodle, rottweiler, mixed-breed, doberman, pig
- sheep
  - cattle, goats, cows, chickens, sheeps, hogs, donkeys, herds, shorthorn, livestock
- november
  - october, december, april, june, february, july, september, january, august, march
- jerusalem
  - tiberias, jaffa, haifa, israel, palestine, nablus, damascus katamon, ramla, safed
- teva
  - pfizer, schering-plough, novartis, astrazeneca, glaxosmithkline, sanofi-aventis, mylan, sanofi, genzyme, pharmacia

- dog
  - cat, dogs, dachshund, rabbit, puppy, poodle, rottweiler, mixed-breed, doberman, pig
- sheep
  - cattle, s, donkeys, herds,
- november
  - octobe ly, september, januar
- jerusalem
  - tiberias, jaffa, haifa, israel, palestine, nablus, damascus katamon, ramla, safed
- teva
  - pfizer, schering-plough, novartis, astrazeneca, glaxosmithkline, sanofi-aventis, mylan, sanofi, genzyme, pharmacia

plug these as alternative inputs to almost any model and get a few points boost in accuracy

My experience with Deep Learning for Language

- With proper tools, easy to produce "innovative" models.
- Not so easy to get good results.
- With Feed-forward nets, hard to beat linear models w/ human engineered feature combinations.
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.
- Semi-sup learning sort-of easy with word-embeddings.

My experience with Deep Learning for Language

- With proper tools, easy to produce "innovative" models.
- Not so easy to get good results.
- With Feed-forward nets, hard to beat linear models w/ human engineered feature combinations.
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.
- Semi-sup learning sort-of easy with word-embeddings.
- **RNNs (in particular LSTMs) are really really cool.**

My experience
with Deep Learning for Language

- With proper tools, easy to produce "innovative" models.
- very ~~Not so~~ easy to get good results.
- With Feed-forward nets, hard to beat linear models w/ human engineered feature combinations.
- On 20-newsgroups, NaiveBayes+TfIdf wins over deep Feed-forward-nets and ConvNets.
- Semi-sup learning sort-of easy with word-embeddings.
- **RNNs (in particular LSTMs) are really really cool.**

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**

Chris Manning
April 2017

Doing stuff with LSTMs

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

Doing stuff with LSTMs

LSTMs are very capable learners

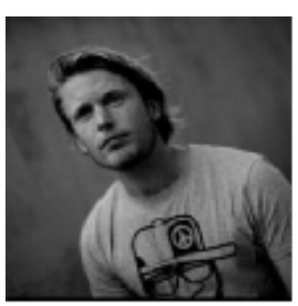
Use them to build stuff

Try to do it in an interesting way

Doing stuff with LSTMs

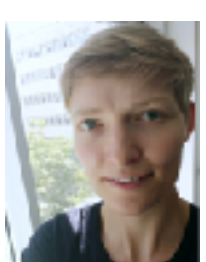
LSTMs are very capable learners

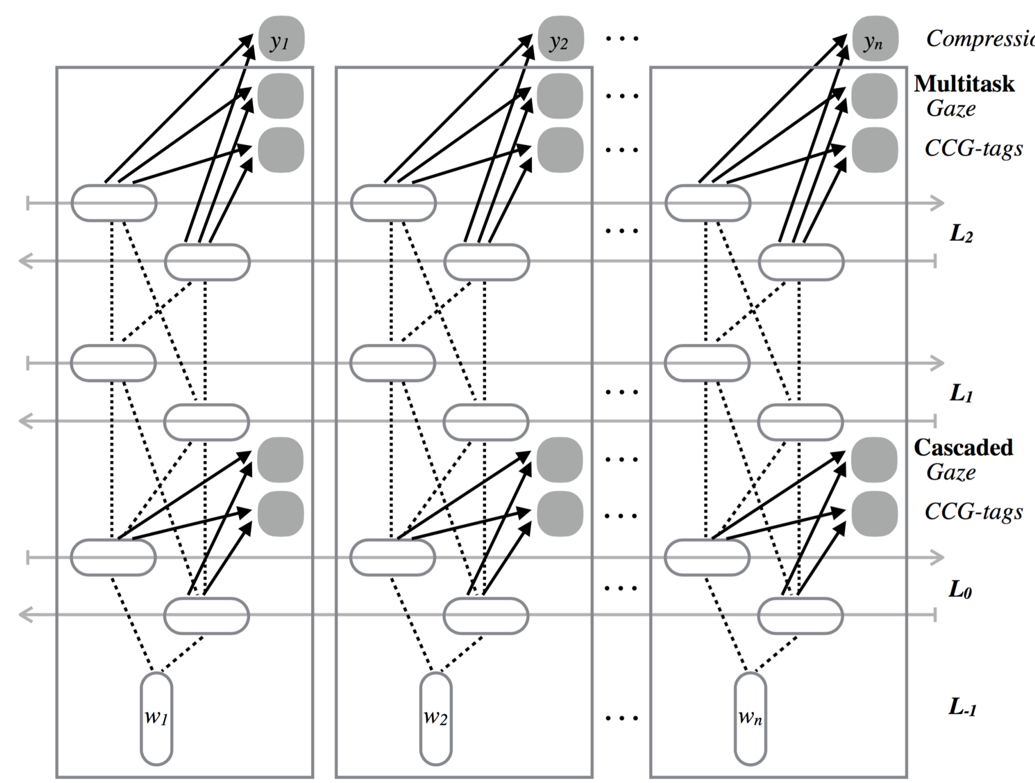
Use them to build stuff



chunking / tagging/ compression

multi-task learning

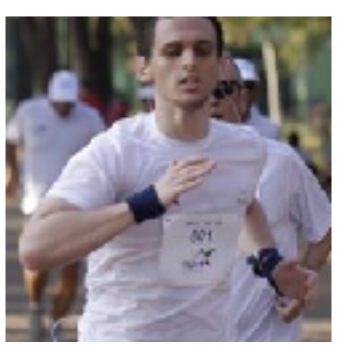




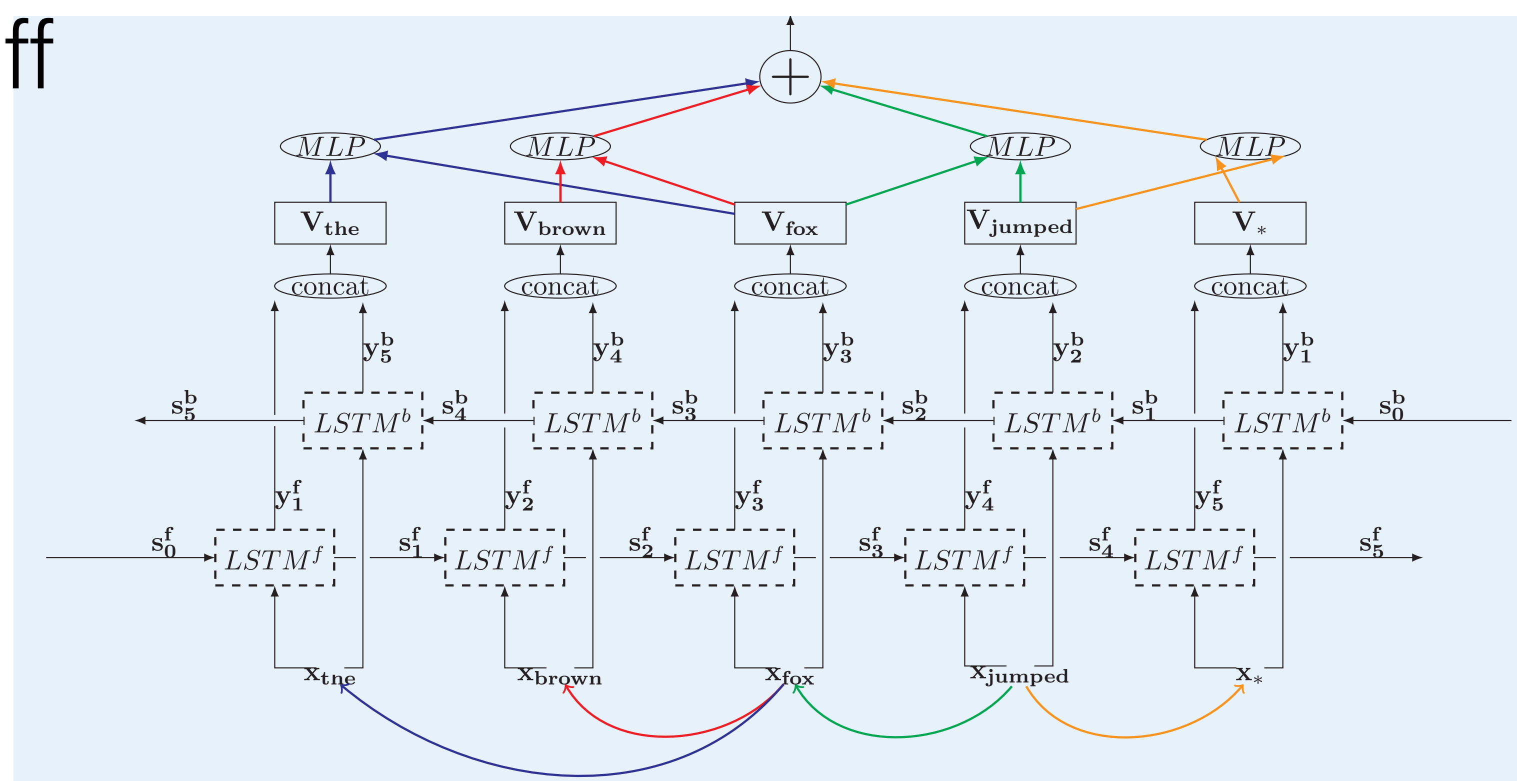
Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

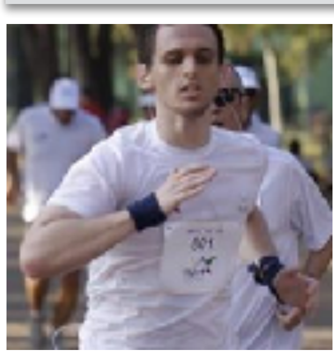


syntactic parsing

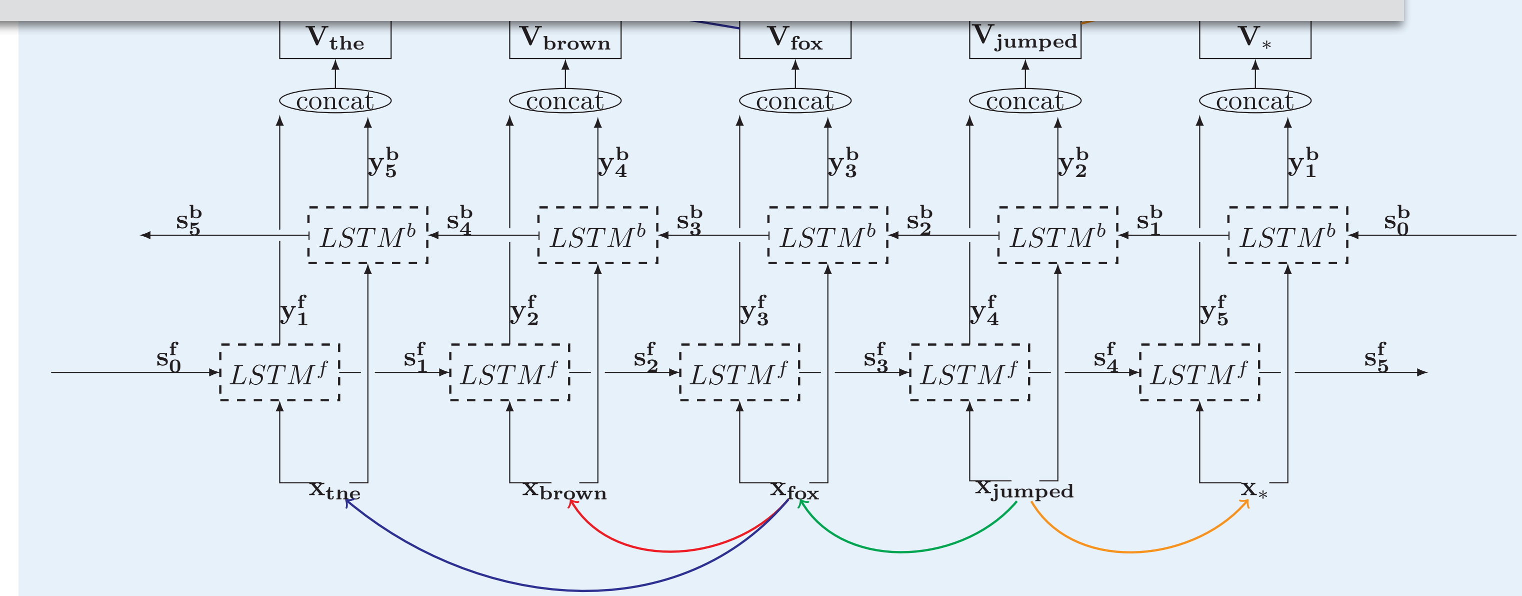


Doing stuff with LSTMs

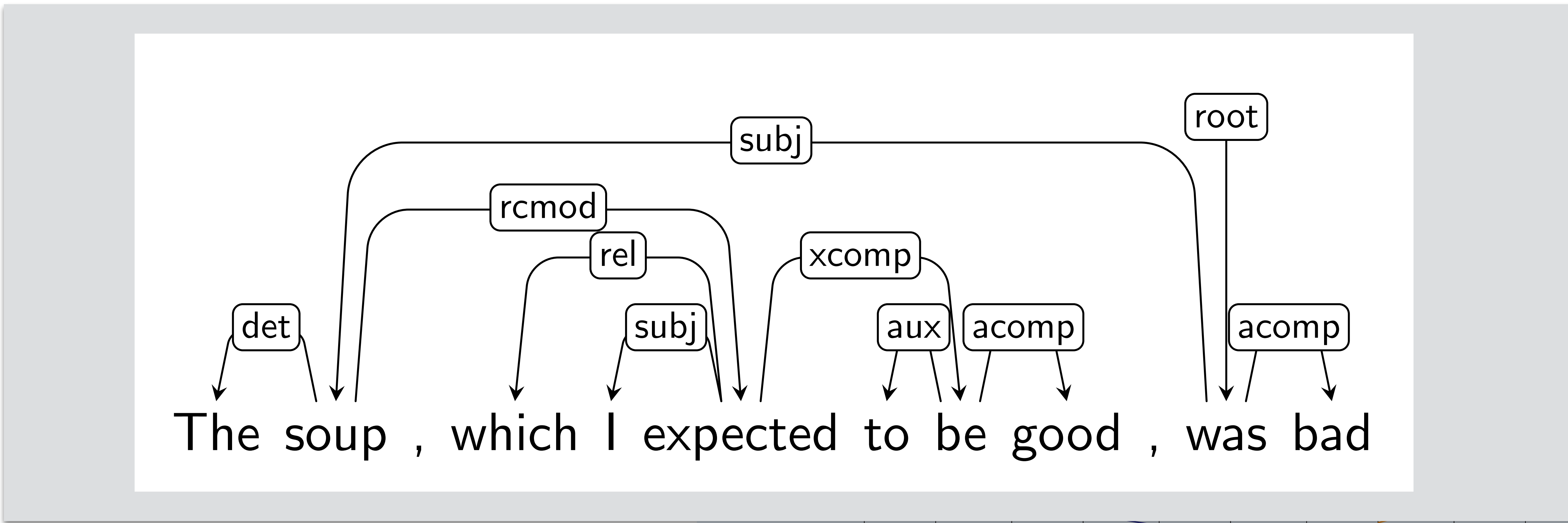
The soup , which I expected to be good , was bad

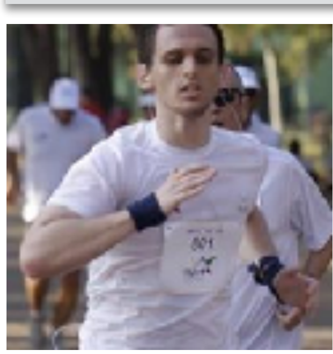


syntactic parsing

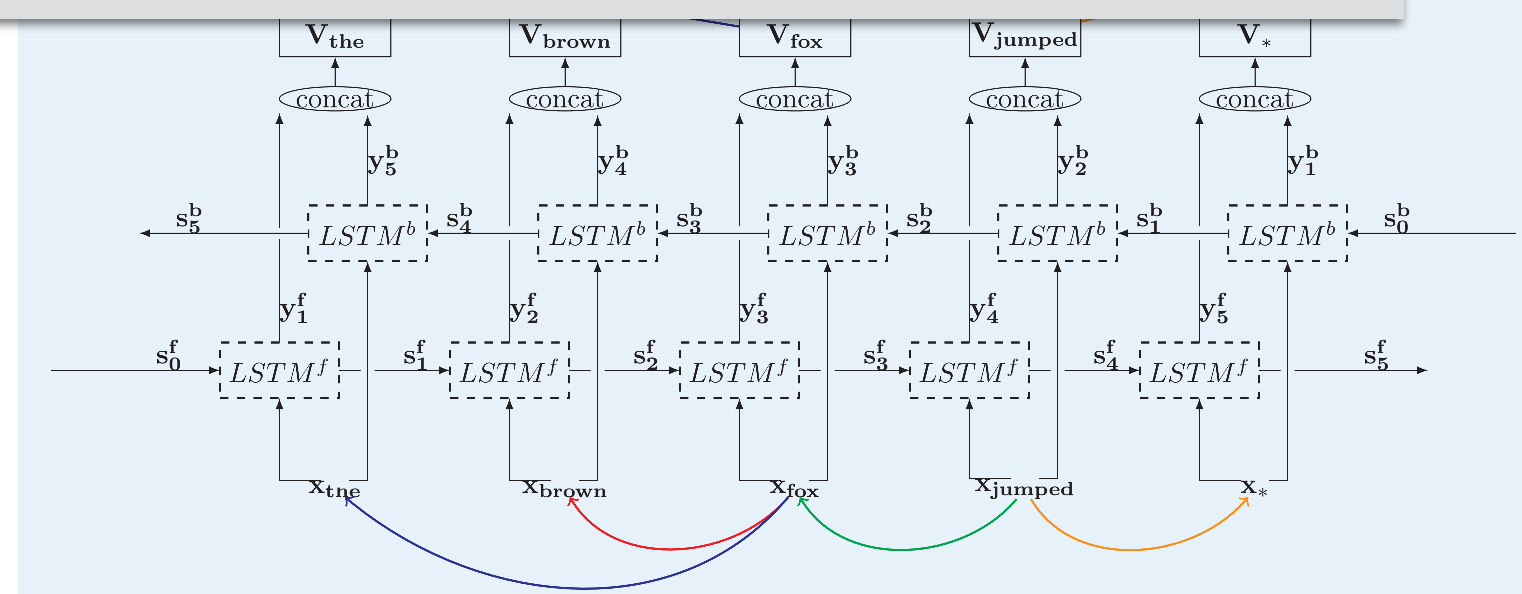


Doing stuff with LSTMs





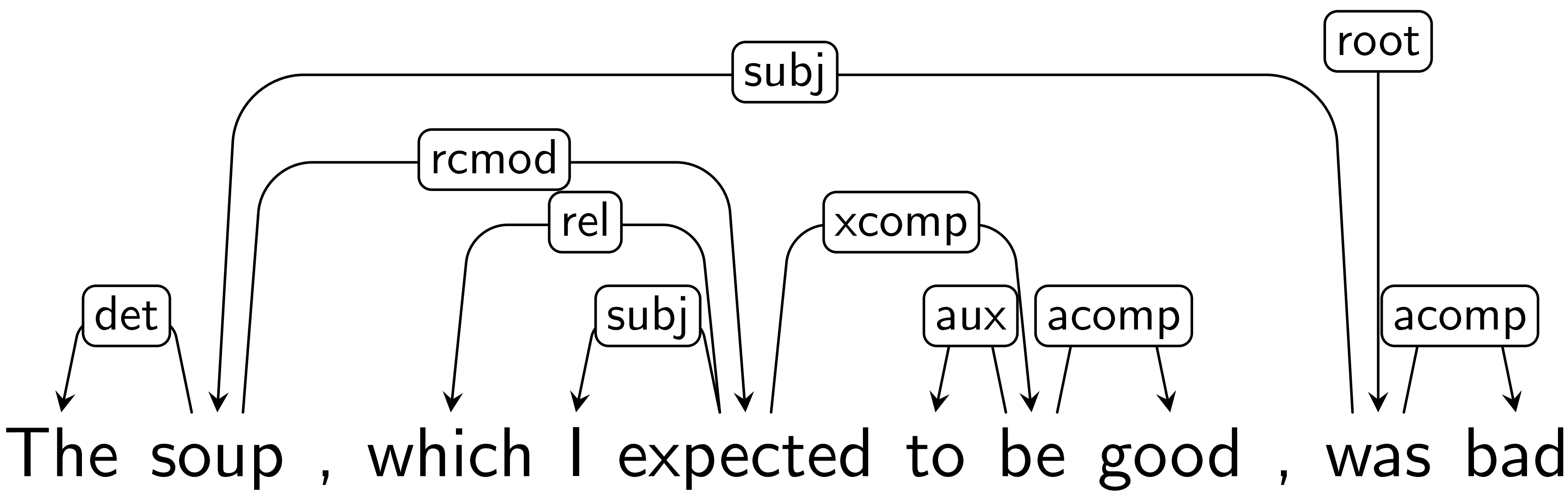
syntactic parsing

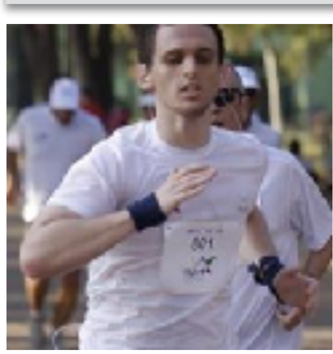


Doing stuff with LSTMs

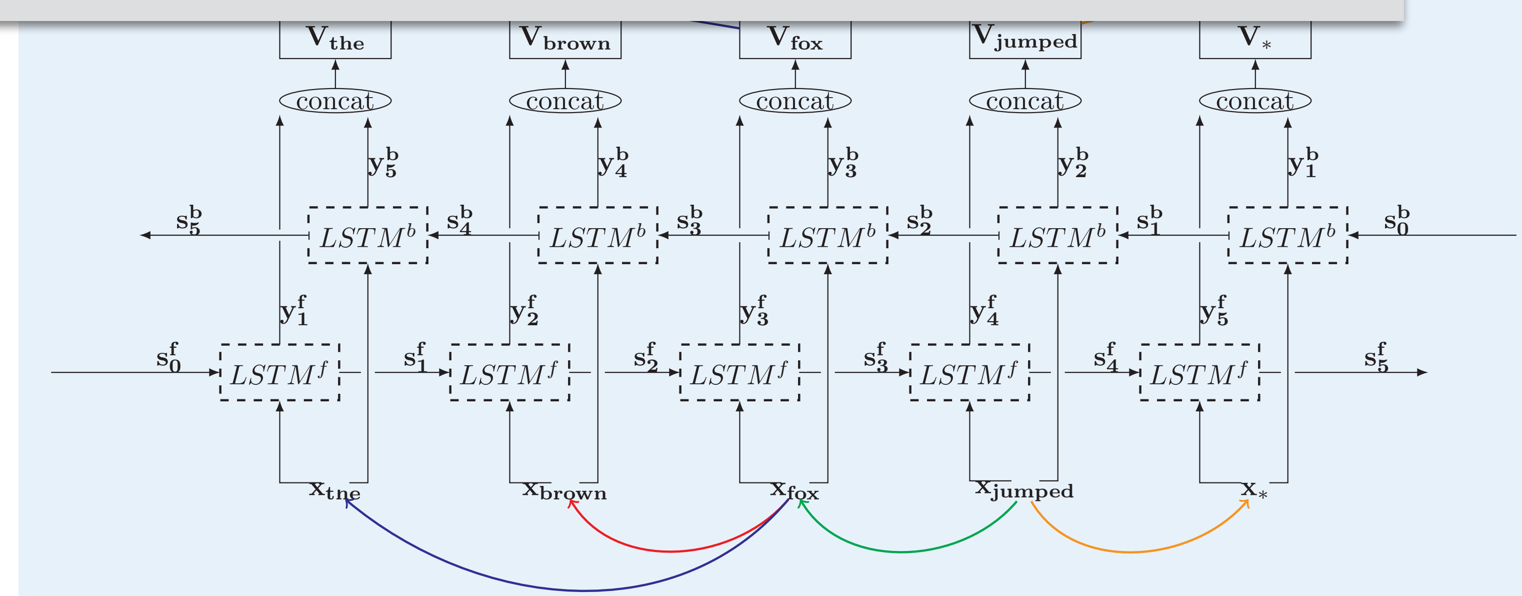
best parser in the world

(now second place, beat by Stanford using same arch)





syntactic parsing

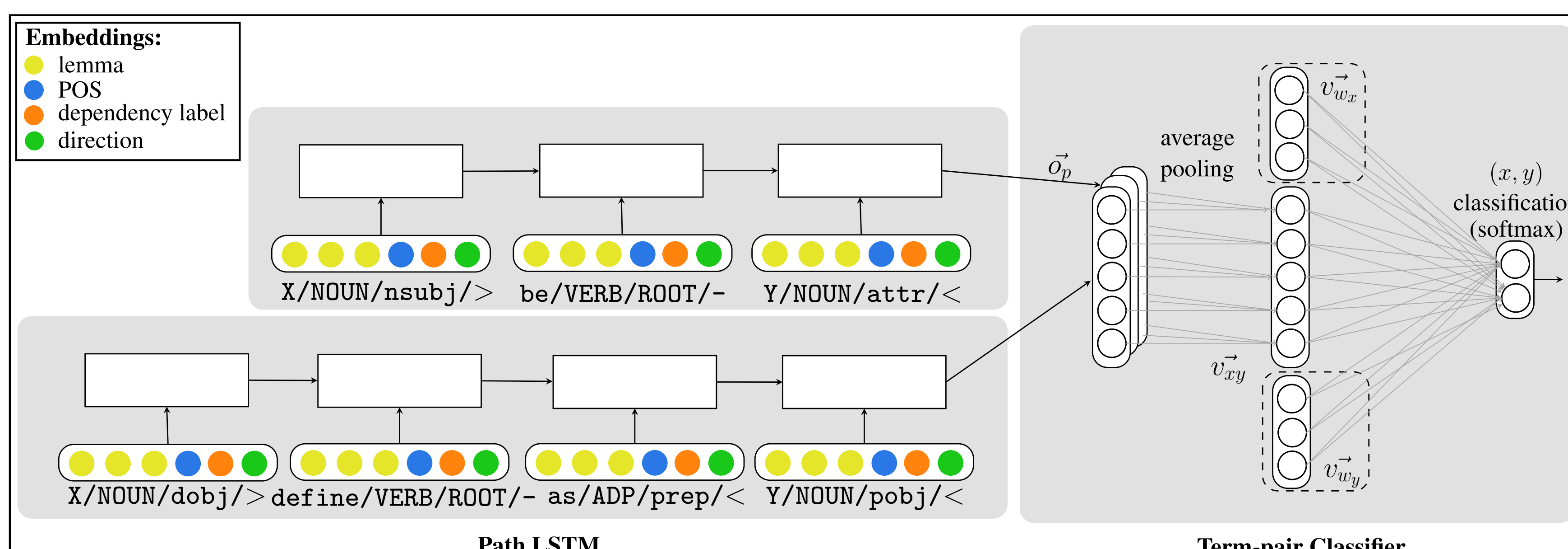


Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

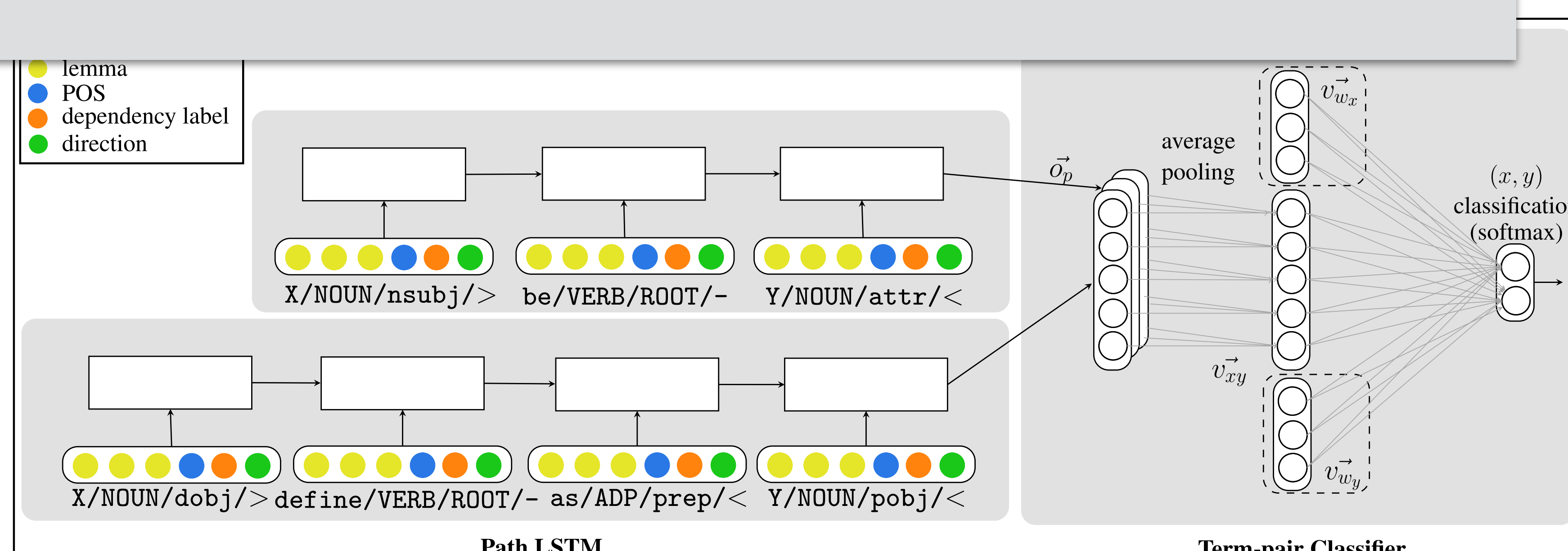
hypernymy detection



Doing stuff with LSTMs

"tuvalu" is a country
"ninjaken" is a weapon
"chlamydophila" is a bacteria

hypernymy detection



Doing stuff with LSTMs

LSTMs are very capable learners

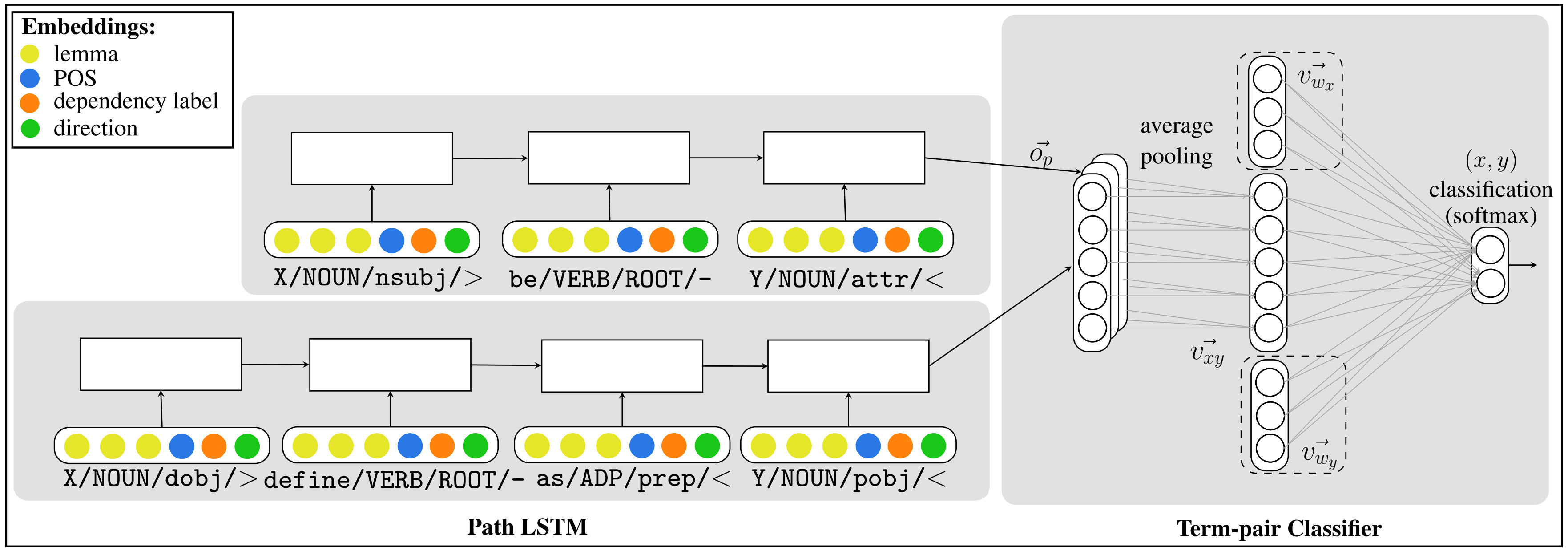
Use them to build stuff

"Outstanding Paper"

"HypeNET"

5/5/5 at ACL!

hypernymy detection

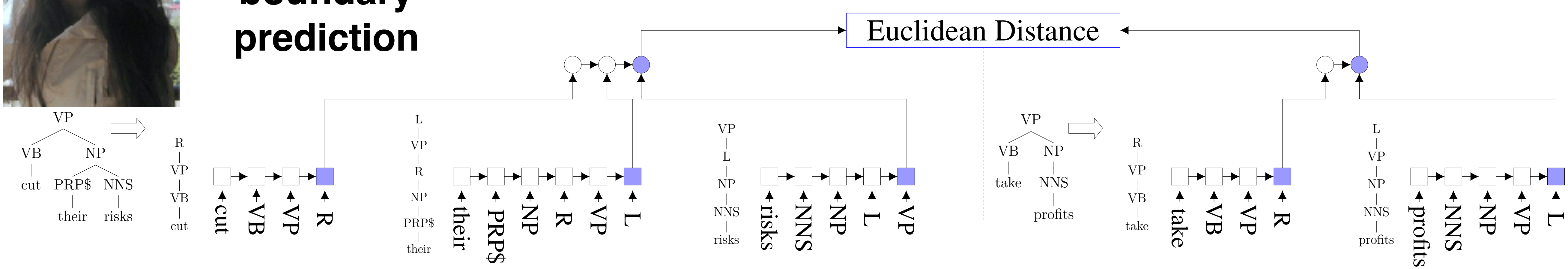


Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

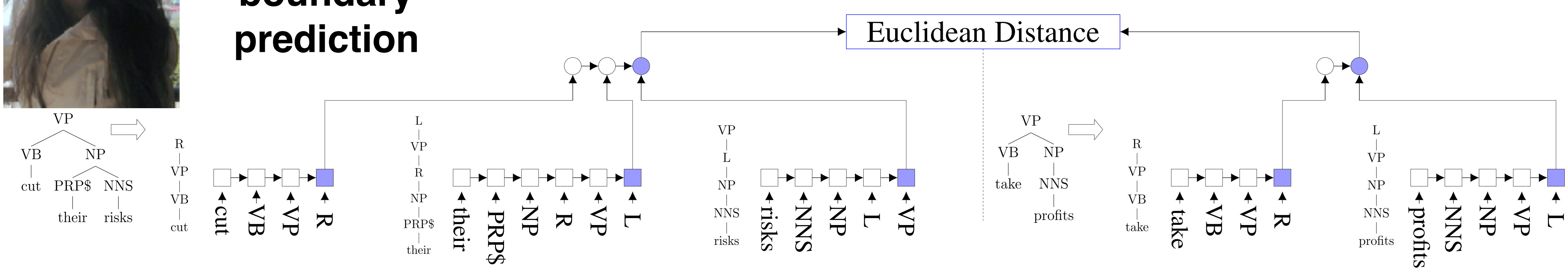
coordination boundary prediction



Doing stuff with LSTMs

he will attend the meeting and present the results on Tuesday

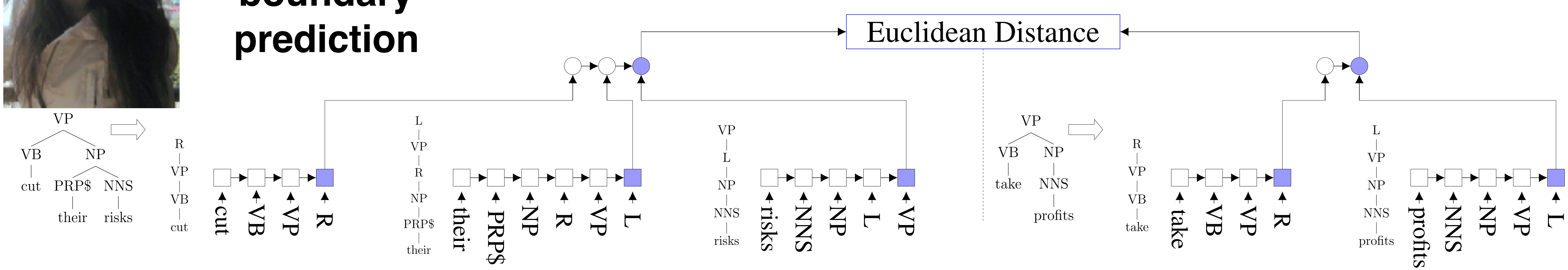
coordination boundary prediction



Doing stuff with LSTMs

he will attend the meeting **and** present the results on Tuesday

coordination boundary prediction



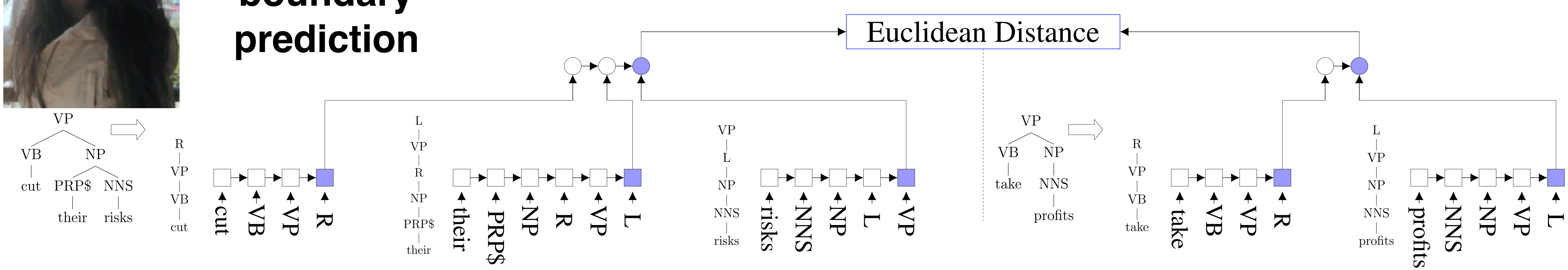
Doing stuff with LSTMs

he will attend the meeting **and** present the results on Tuesday

he will attend the meeting on Tuesday

he will present the results on Tuesday

coordination boundary prediction

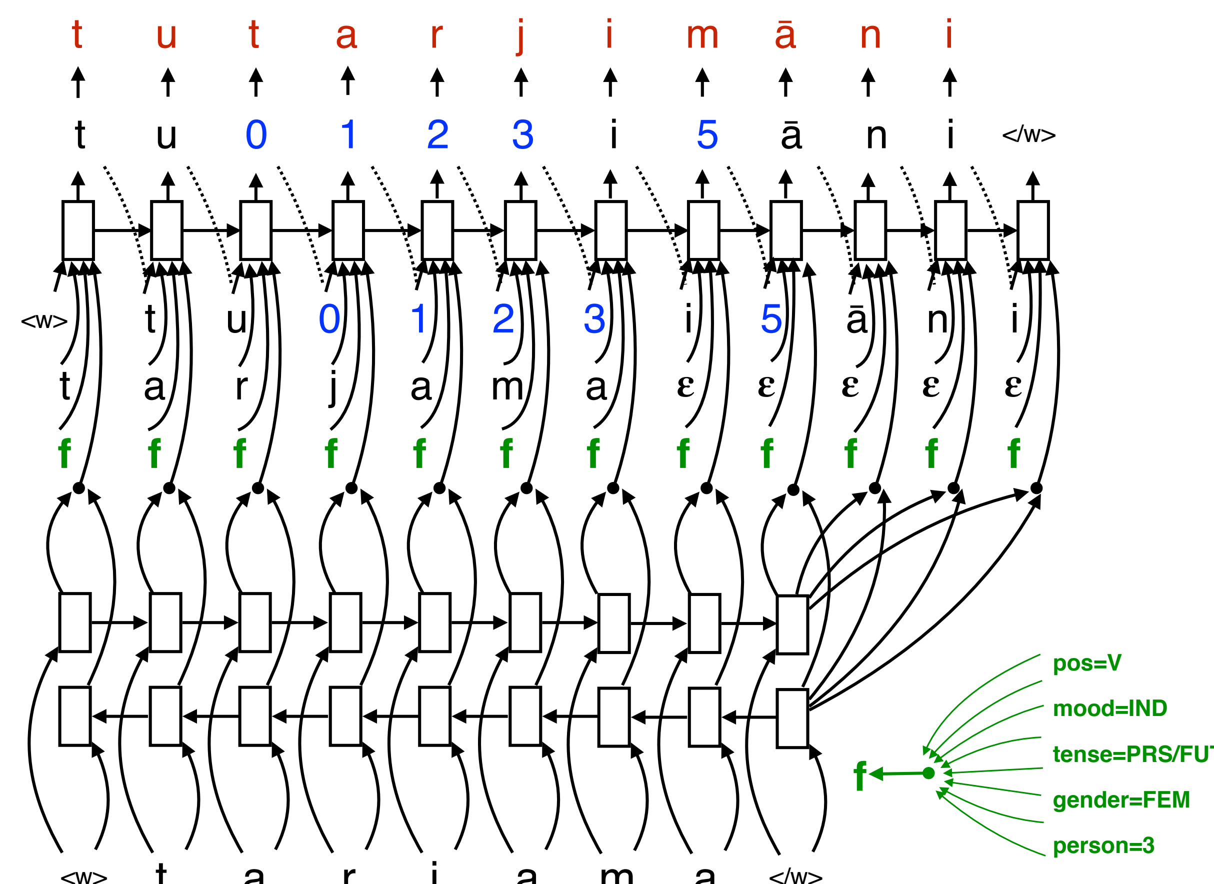


Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

morphological reinflection

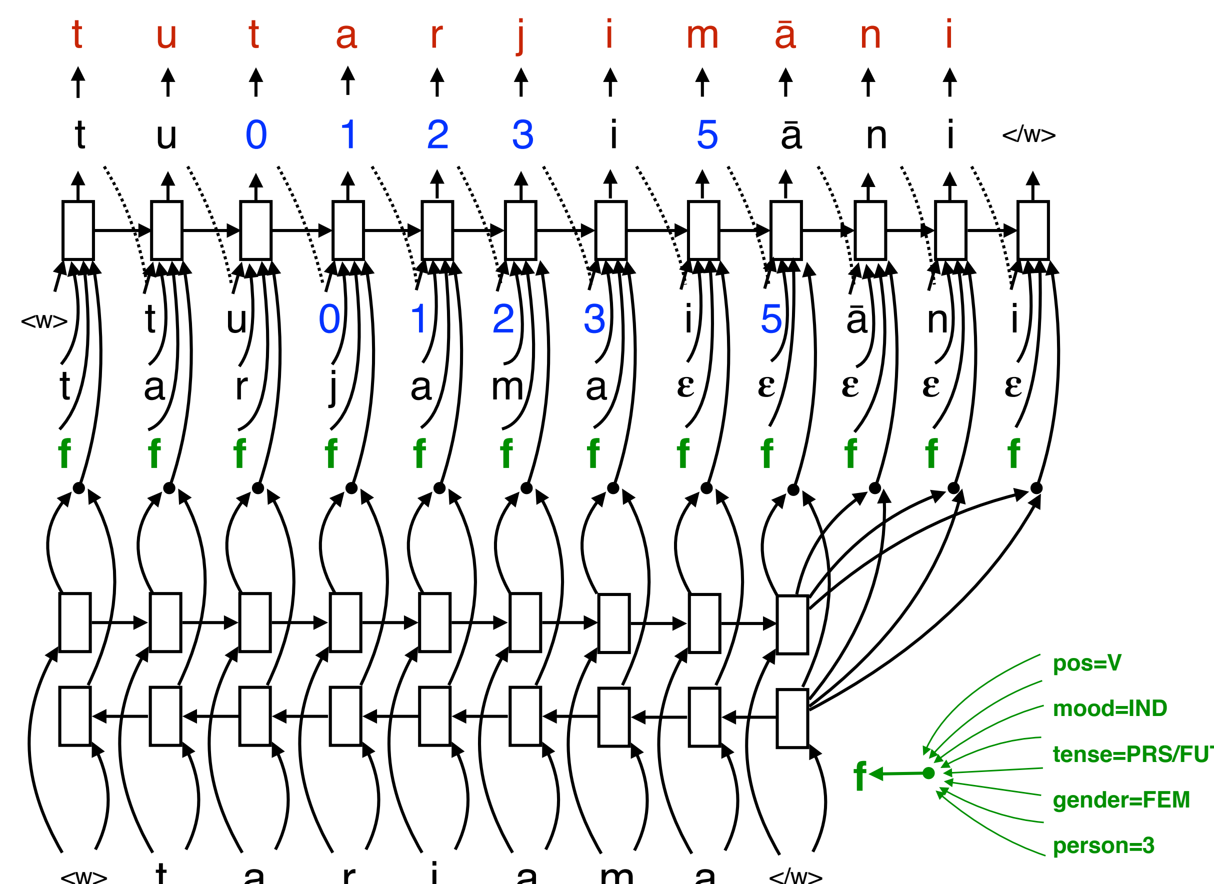


Doing stuff with LSTMs

להטות -VerbInf +Noun,Plural → הטיות

Use them to build stuff

morphological reinflection

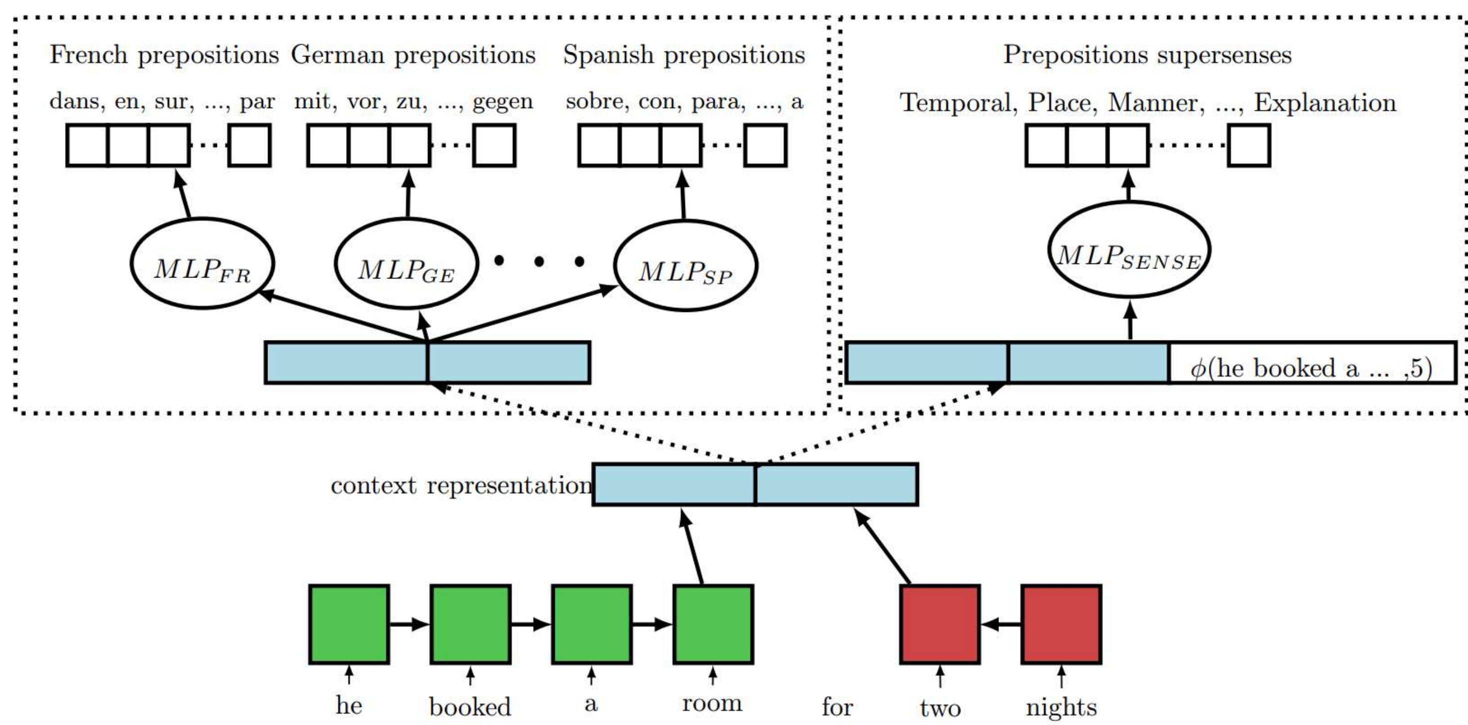


Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

preposition sense disambiguation + semisup on multilingual data

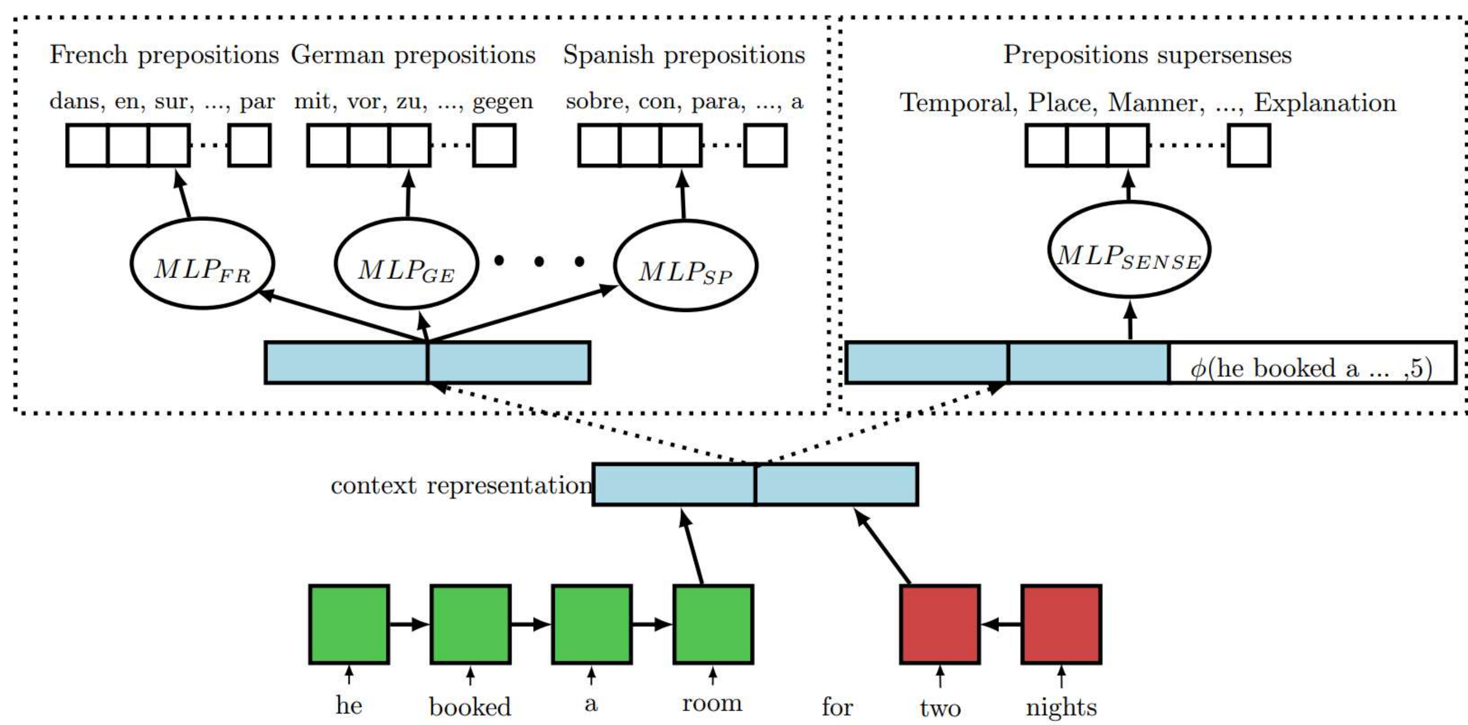


Doing stuff with LSTMs

I met him **for** lunch → Purpose
He paid **for** me → Beneficiary
We sat there **for** hours → Duration

Use them to build stuff

preposition sense disambiguation + semisup on multilingual data

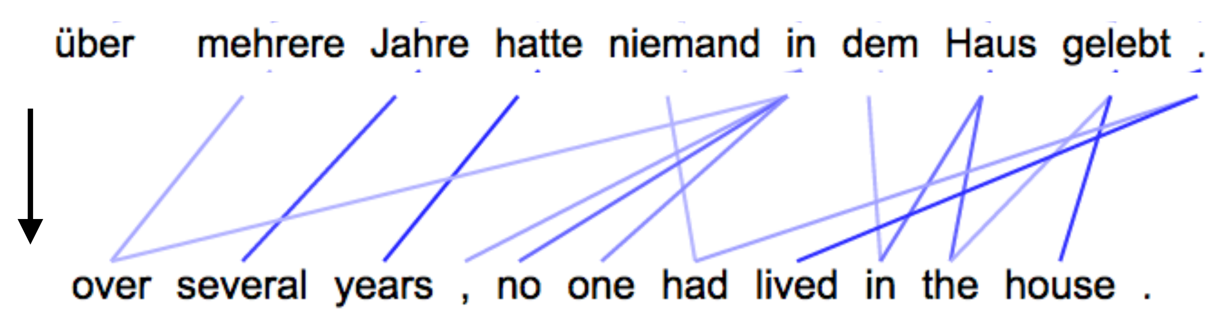


Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

syntax based machine translation

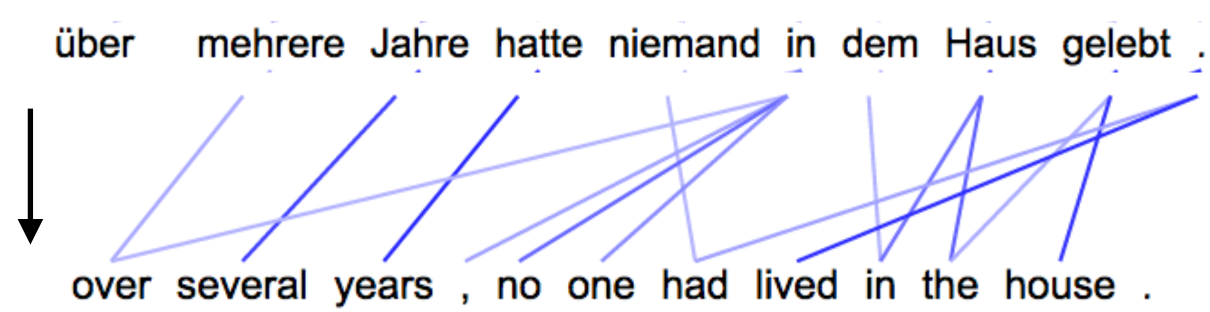


Doing stuff with LSTMs

German → English

Use them to build stuff

syntax based machine translation

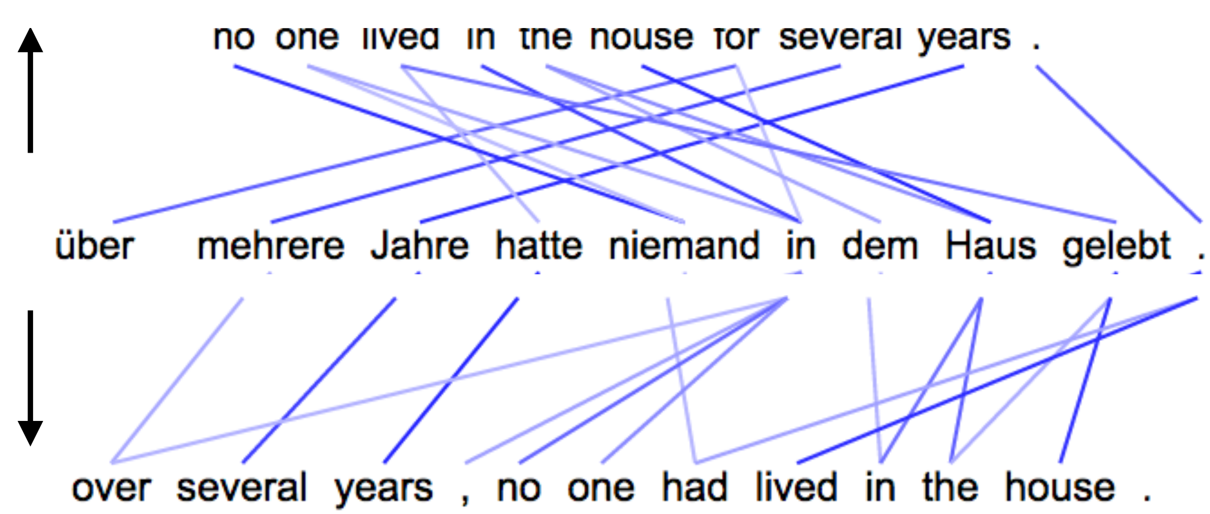


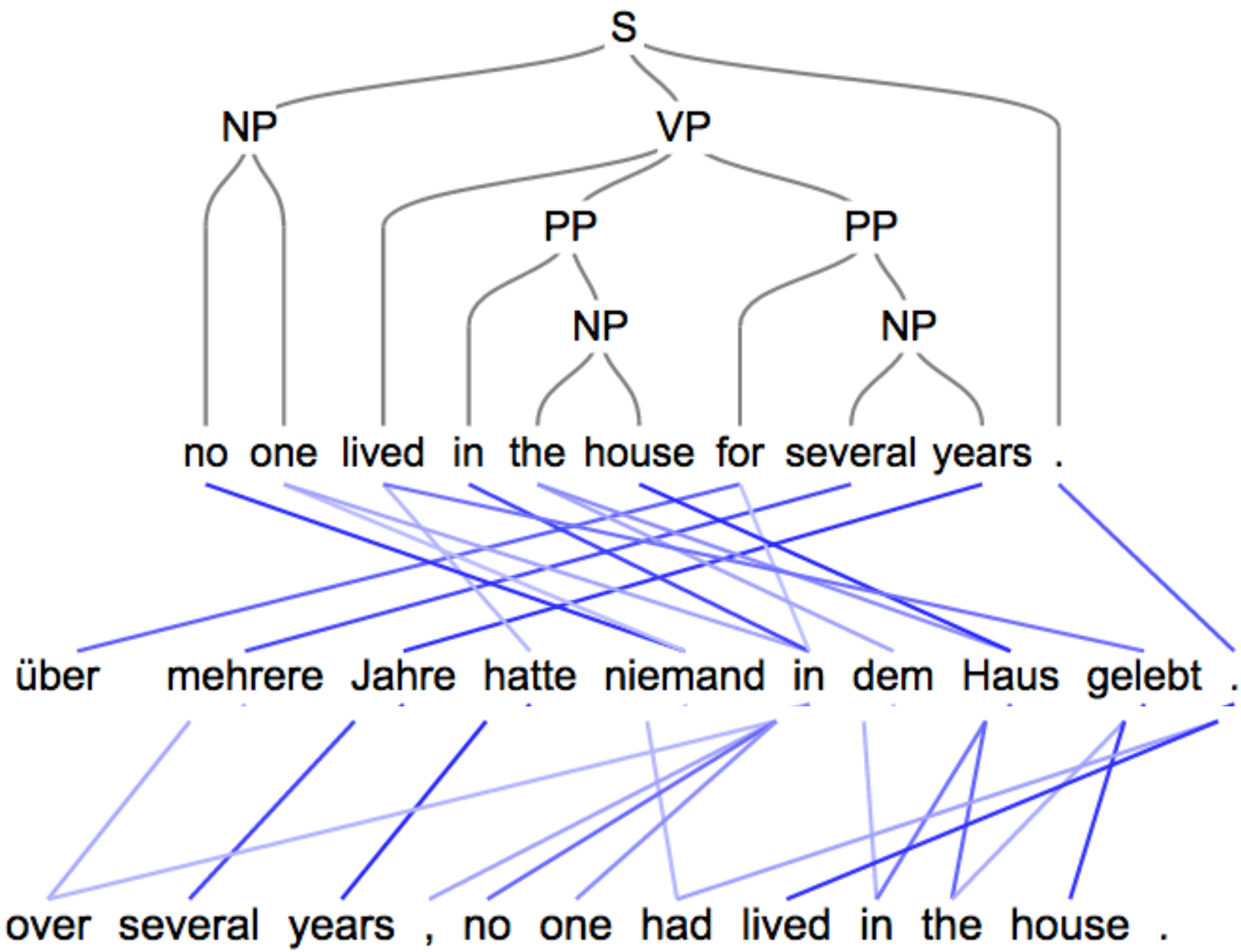
Doing stuff with LSTMs

German → English

Use them to build stuff

syntax based machine translation




S
NP
VP
PP
PP
NP
NP
no one lived in the house for several years .
über mehrere Jahre hatte niemand in dem Haus gelebt .
over several years , no one had lived in the house .

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

text generation with style

Doing stuff with LSTMs

Parameter	Value
Theme	Plot
Sentiment	Positive
Writer Type	Audience
Subjective	False
Length	11-20 words
Descriptive	False

text generation with style

Doing stuff with LSTMs

Parameter	Value
Theme	Plot
Sentiment	Positive
Writer Type	Audience
Subjective	False
Length	11-20 words
Descriptive	False

- "It 's a touching story that will keep you on the edge of your seat the whole time ! ! !"
- "The story was not quite as good as the first one but it had a pretty good twist ending."

text generation with style

Doing stuff with LSTMs

Parameter	Value
Theme	Other
Sentiment	Negative
Writer Type	Audience
Subjective	True
Length	11-20 words
Descriptive	True

- "My biggest problem with the whole movie though is that there is nothing new or original or great in this film."
- "Ultimately, I can honestly say that this movie is full of stupid stupid and stupid stupid stupid stupid stupid."

text generation with style

Doing stuff with LSTMs

LSTMs are very capable learners

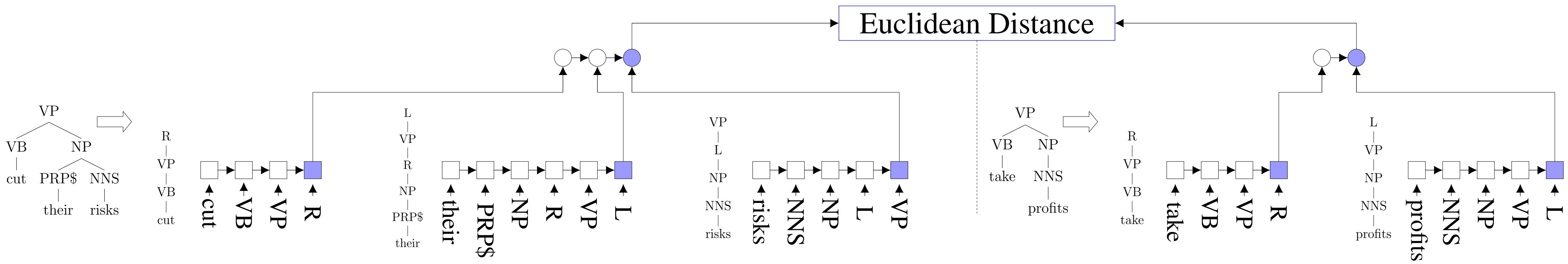
Use them to build stuff

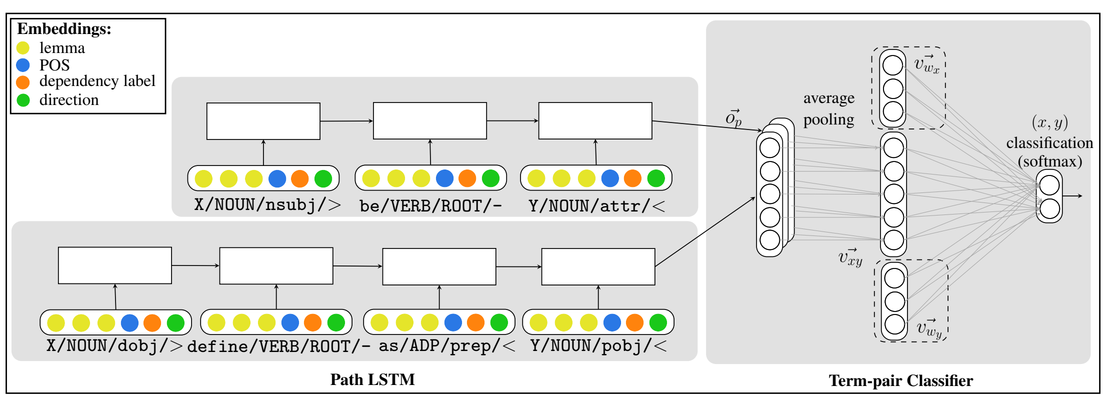
strong results

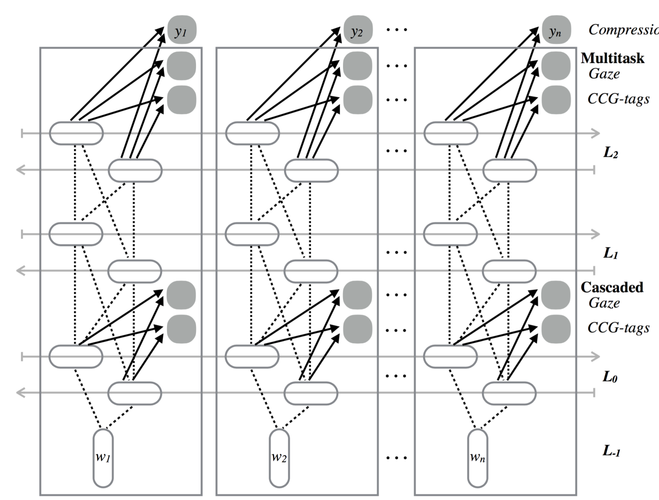
make reviewers happy

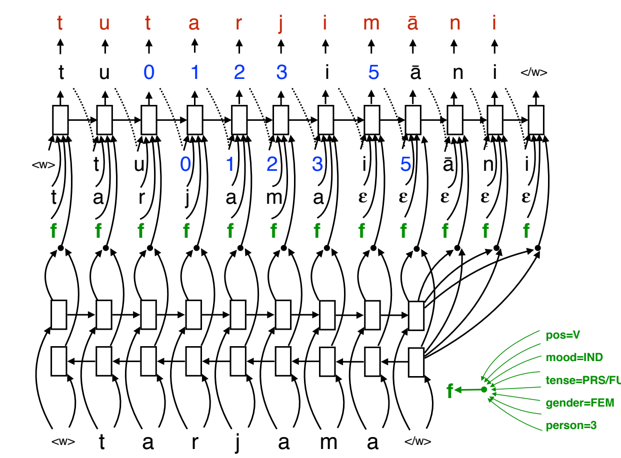
publish many papers

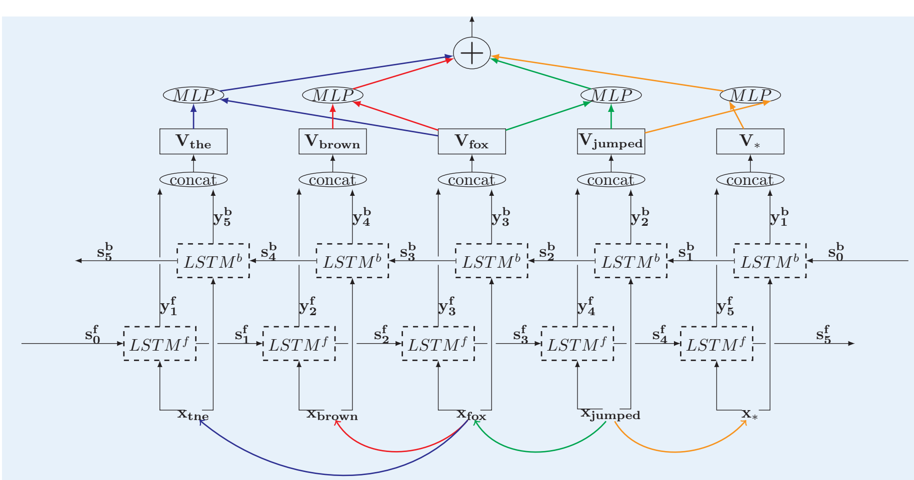
Doing stuff with LSTMs

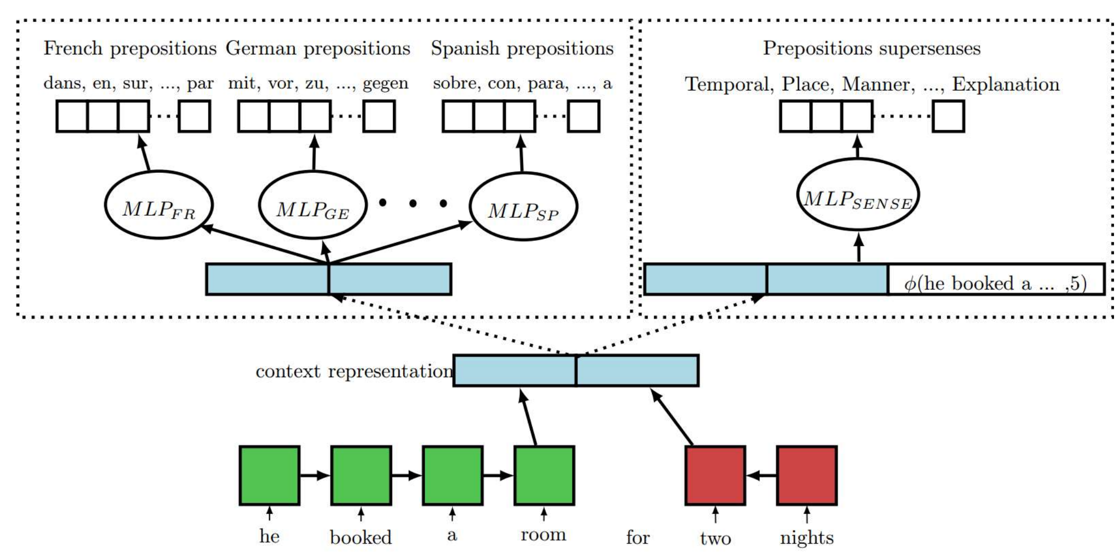












Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

strong results

make reviewers happy

publish many papers

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

strong results

make reviewers happy

publish many papers

build tools to build stuff

∂y/net

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

strong results

make reviewers happy

publish many papers

build tools to build stuff

build stuff faster

help others build stuff

publish more papers

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

strong results

make reviewers happy

publish many papers

build tools to build stuff

build stuff faster

help others build stuff

publish more papers

Try to understand them

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

Try to understand them

scratching the surface

reviewers don't care much

I find it really interesting

Poking at ~~Doing stuff with~~ LSTMs

Yoav Goldberg

Dec 2017

BIU NLP

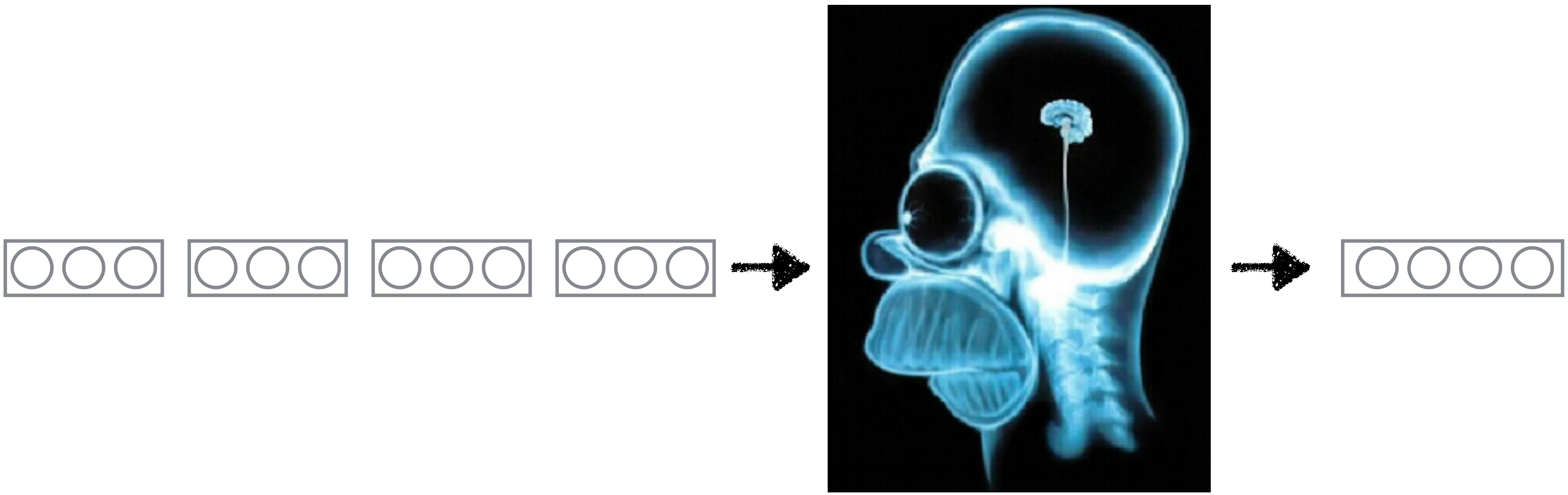
Bar-Ilan University
אוניברסיטת בר-אילן

Agenda

- Inspecting vector representations of sentences
- LSTMs and hierarchical syntax
- Extracting FSAs from RNNs

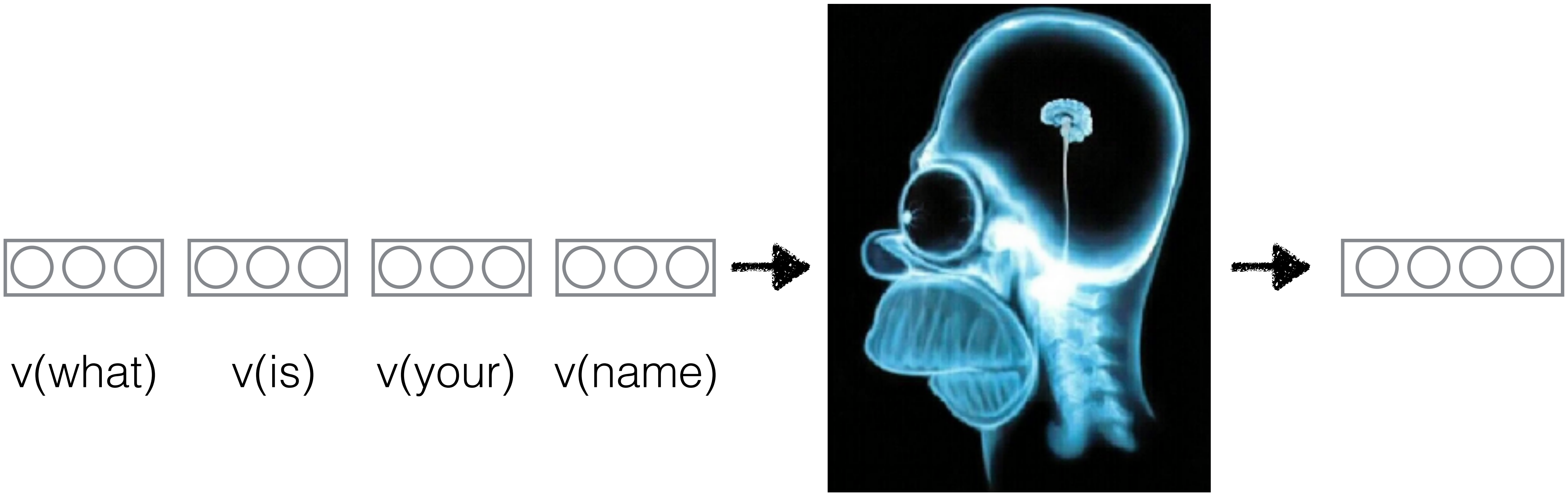
brief intro to RNNs

Recurrent Neural Networks



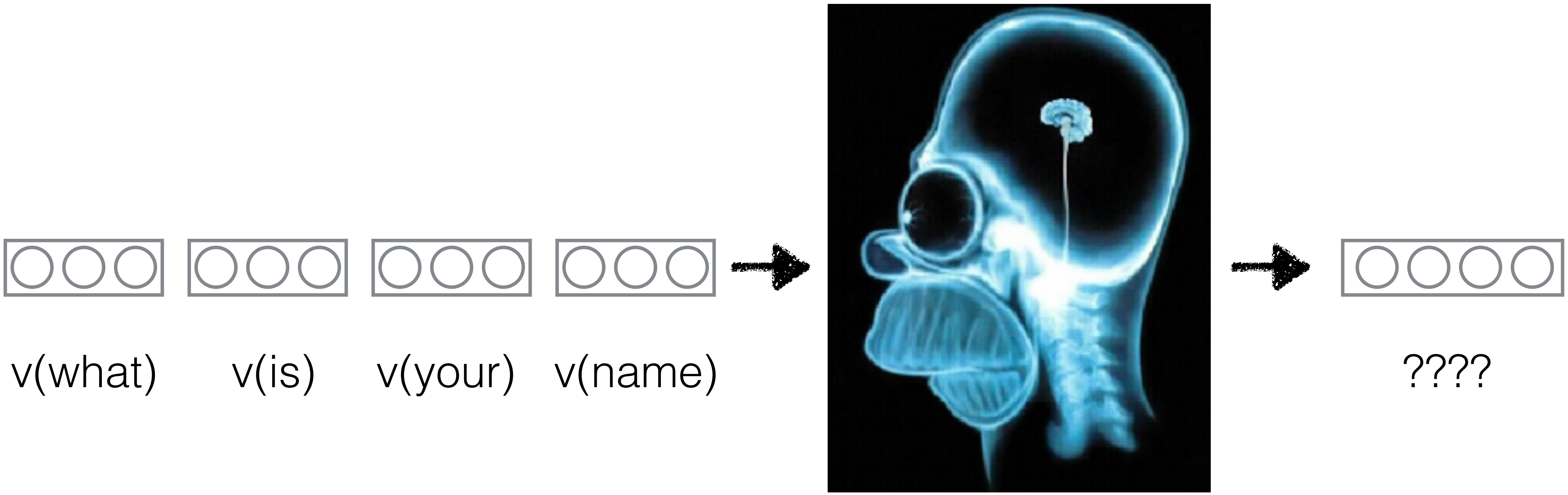
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

Recurrent Neural Networks



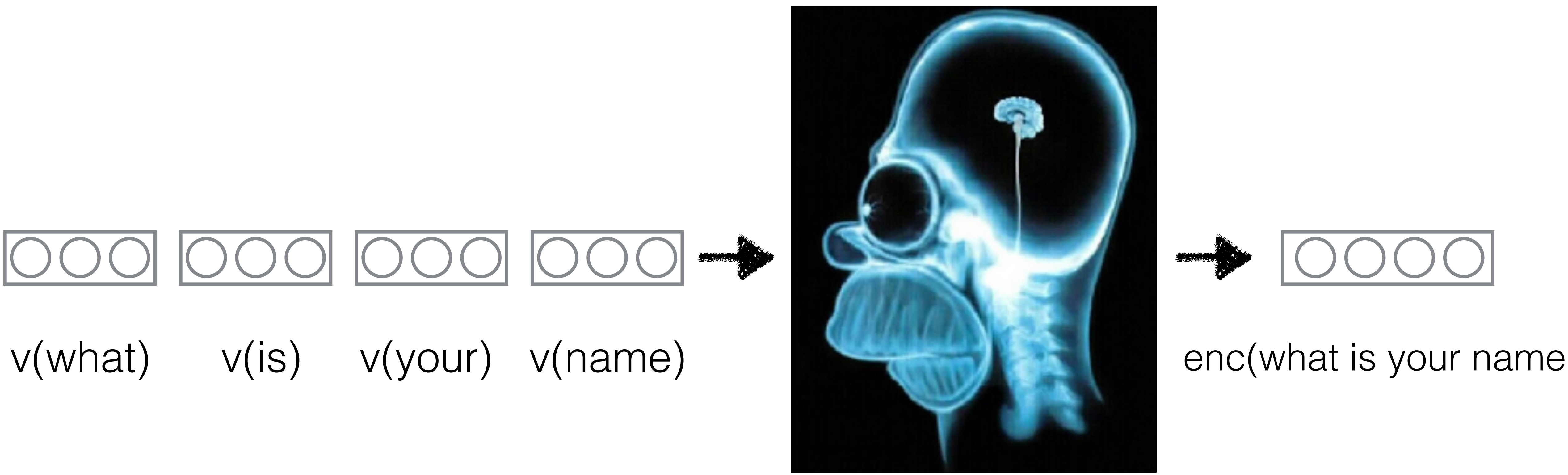
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

Recurrent Neural Networks



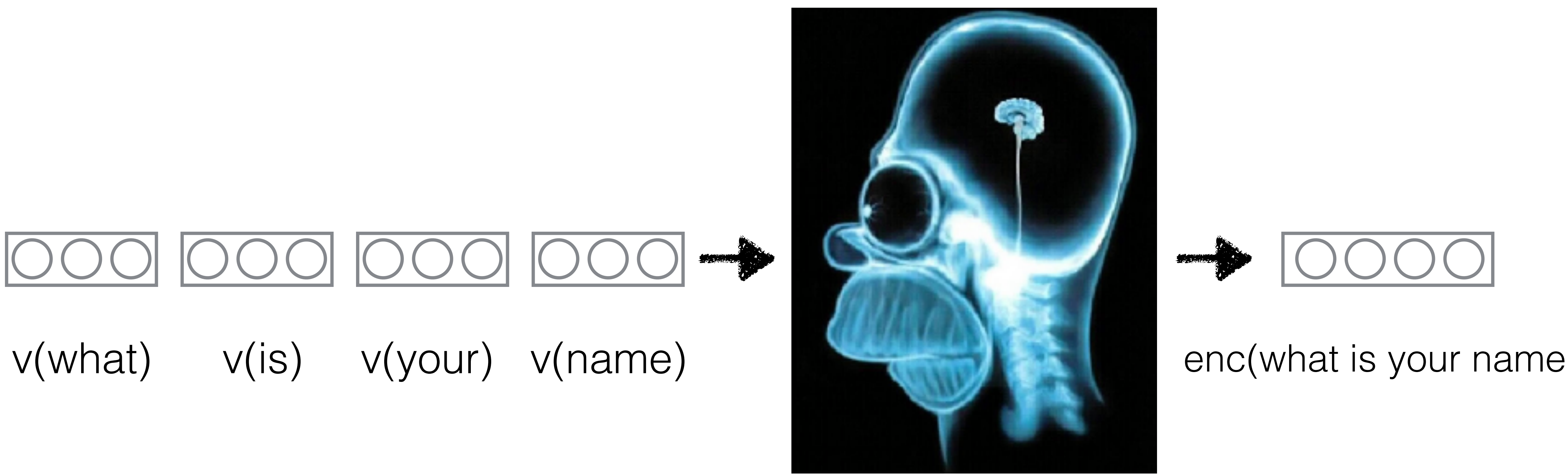
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

Recurrent Neural Networks



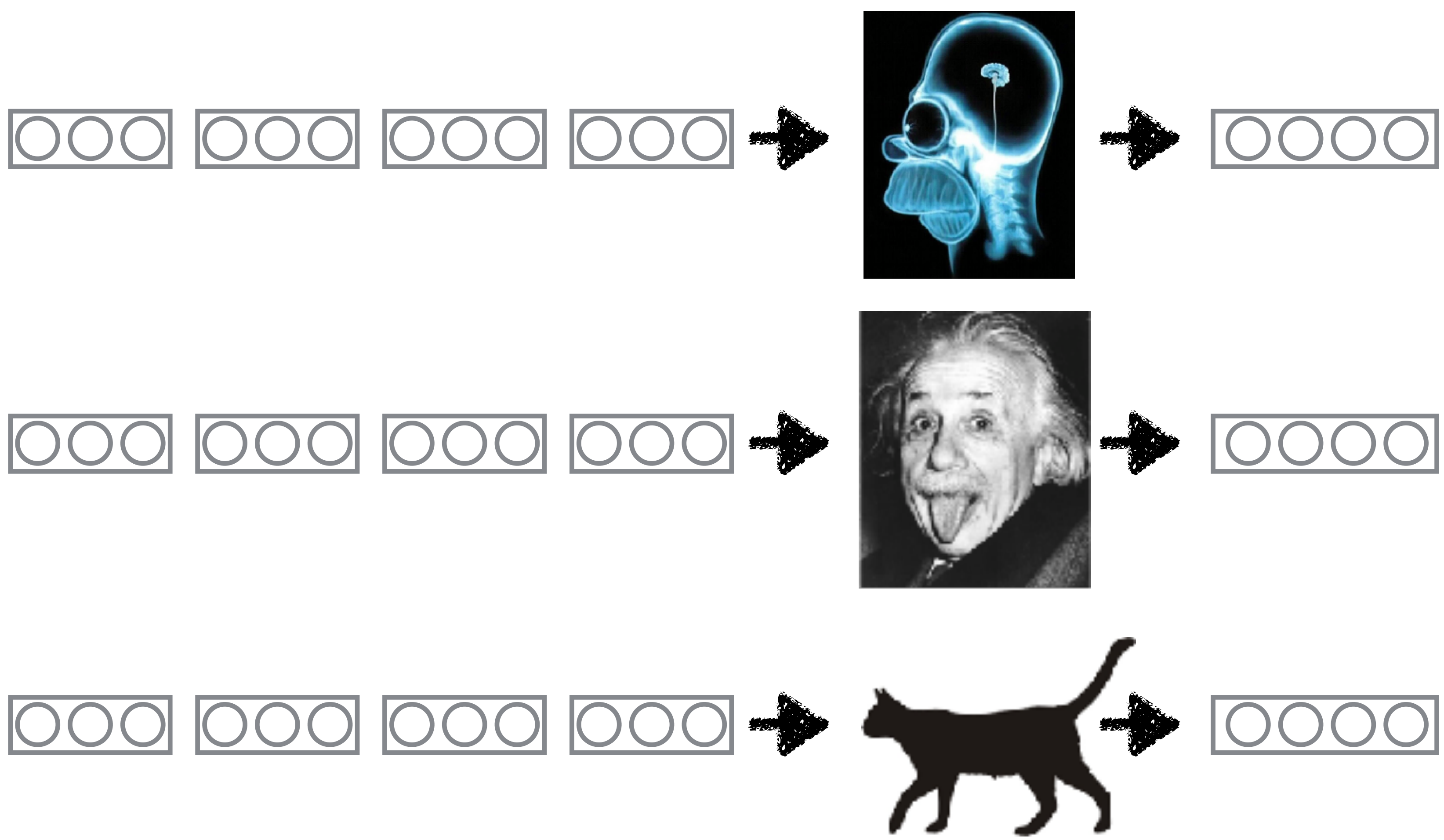
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

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).
- We'll focus on the interface level.

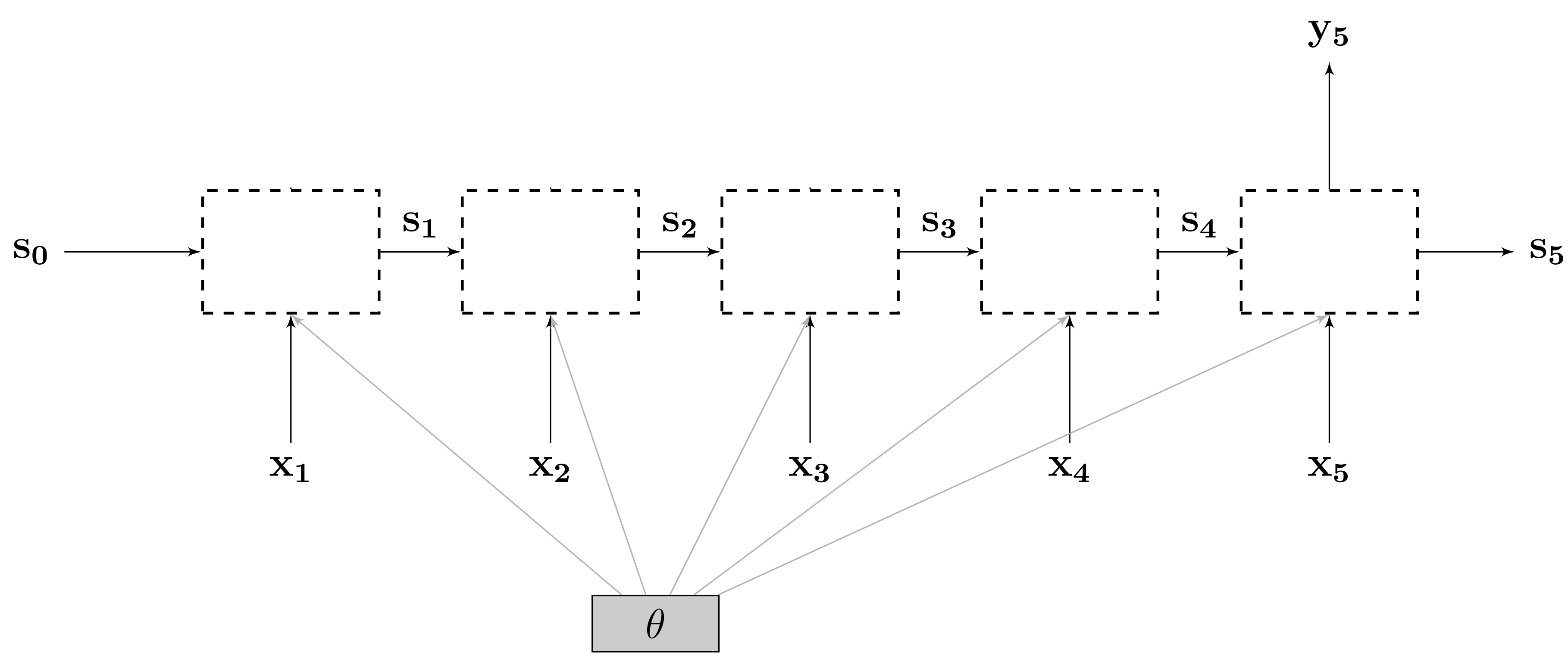
Recurrent Neural Networks

$$RNN(\mathbf{s_0}, \mathbf{x_{1:n}}) = \mathbf{s_n}, \mathbf{y_n}$$

$$\mathbf{x_i} \in \mathbb{R}^{d_{in}},\ \mathbf{y_i} \in \mathbb{R}^{d_{out}},\ \mathbf{s_i} \in \mathbb{R}^{f(d_{out})}$$

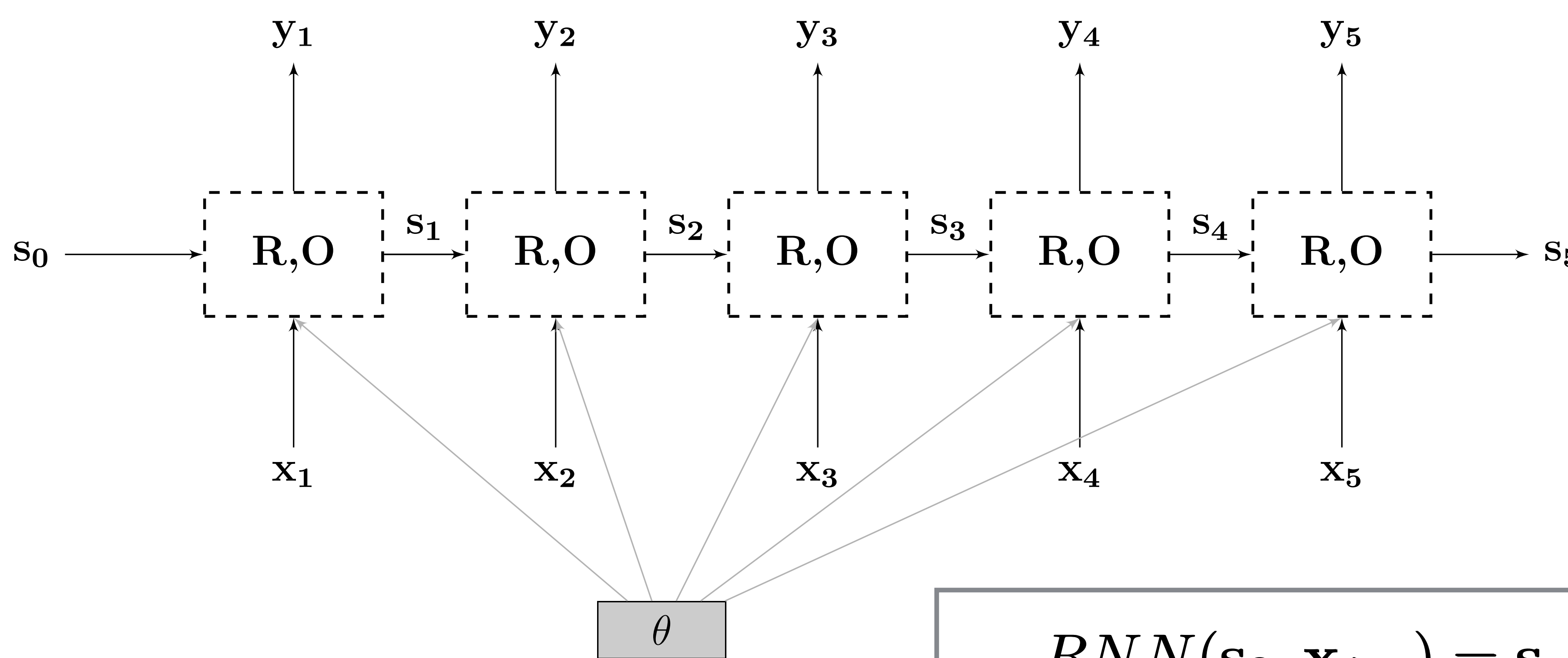
- Very strong models of sequential data.
- **Trainable** function from *n* vectors to a single* vector.

Recurrent Neural Networks



- Input vectors $\mathbf{x_{1:i}}$, output vector $\mathbf{y_i}$
- The output vector $\mathbf{y_i}$ depends on **all** inputs $\mathbf{x_{1:i}}$

Recurrent Neural Networks



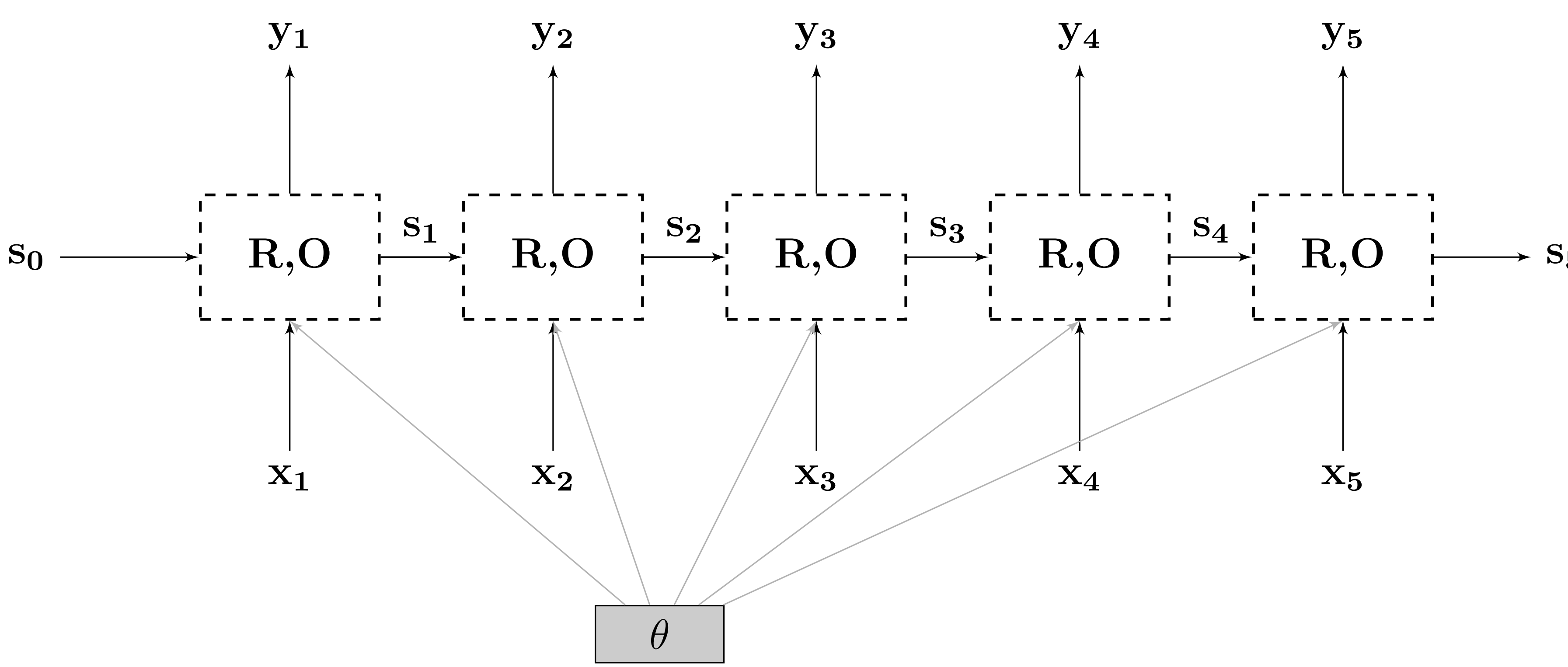
$$RNN(\mathbf{s_0}, \mathbf{x_{1:n}}) = \mathbf{s_n}, \mathbf{y_n}$$
$$\mathbf{s_i} = R(\mathbf{s_{i-1}}, \mathbf{x_i})$$
$$\mathbf{y_i} = O(\mathbf{s_i})$$

$$\mathbf{x_i} \in \mathbb{R}^{d_{in}},\ \mathbf{y_i} \in \mathbb{R}^{d_{out}},\ \mathbf{s_i} \in \mathbb{R}^{f(d_{out})}$$

- Recursively defined.
- There's a vector $\mathbf{y_i}$ for every prefix $\mathbf{x_{1:i}}$

Recurrent Neural Networks

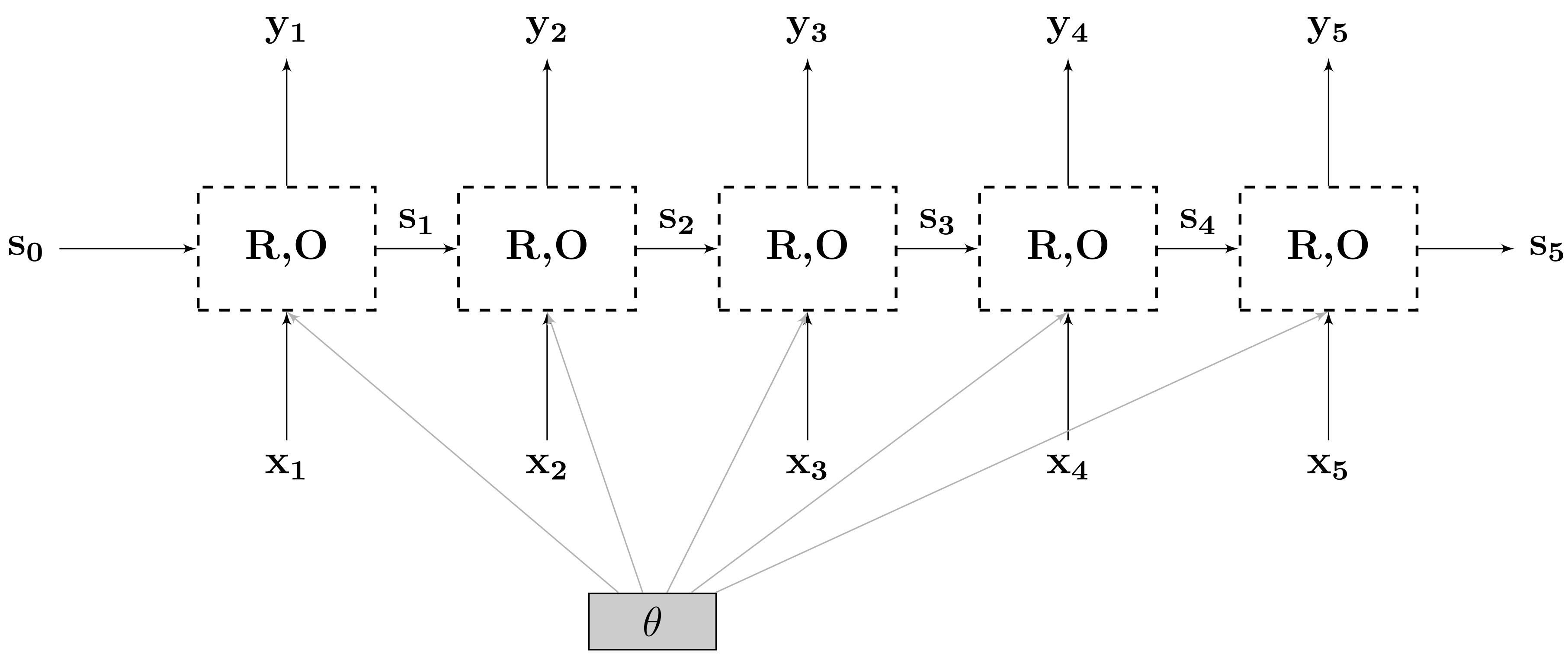
- What are the vectors $\mathbf{y_i}$ good for?



- On their own? **nothing**.

Recurrent Neural Networks

- What are the vectors $\mathbf{y_i}$ good for?

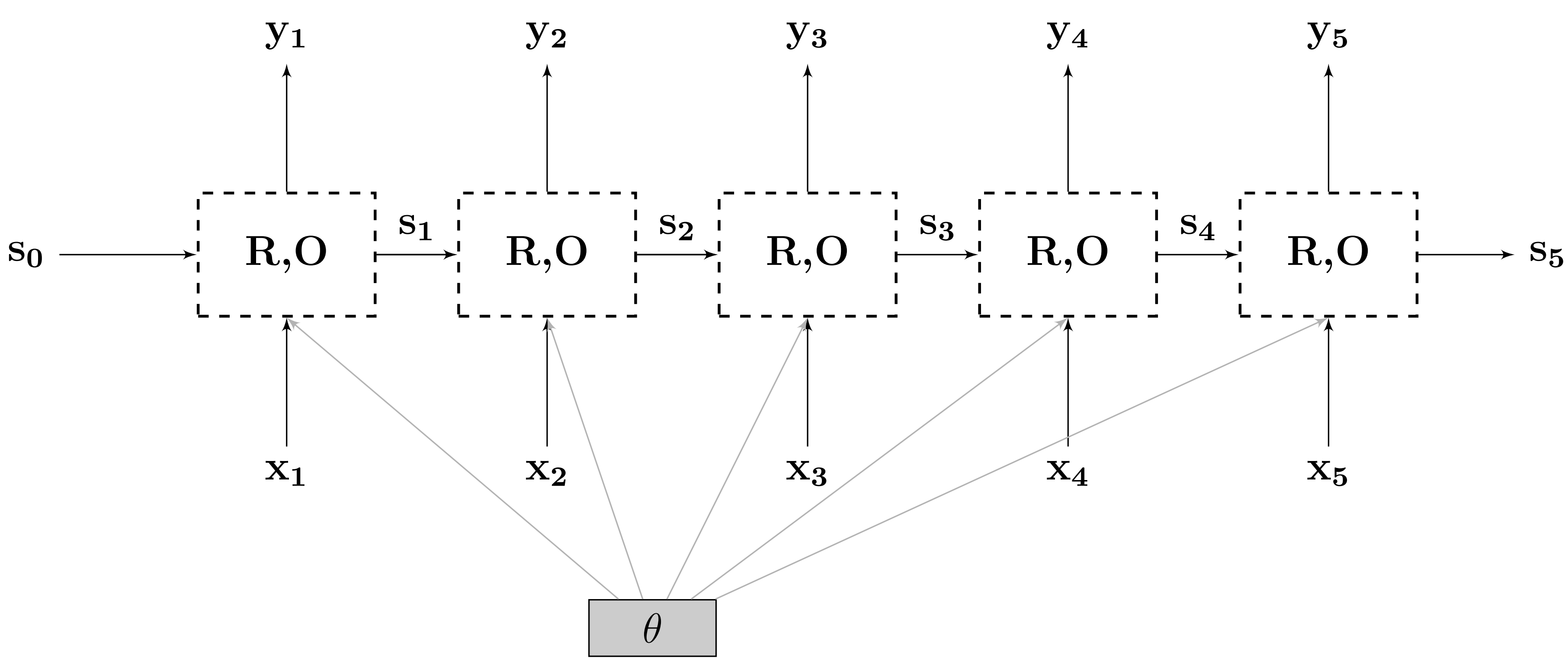


- On their own? **nothing**.

- **But we can train them.**

Recurrent Neural Networks

- What are the vectors $\mathbf{y_i}$ good for?



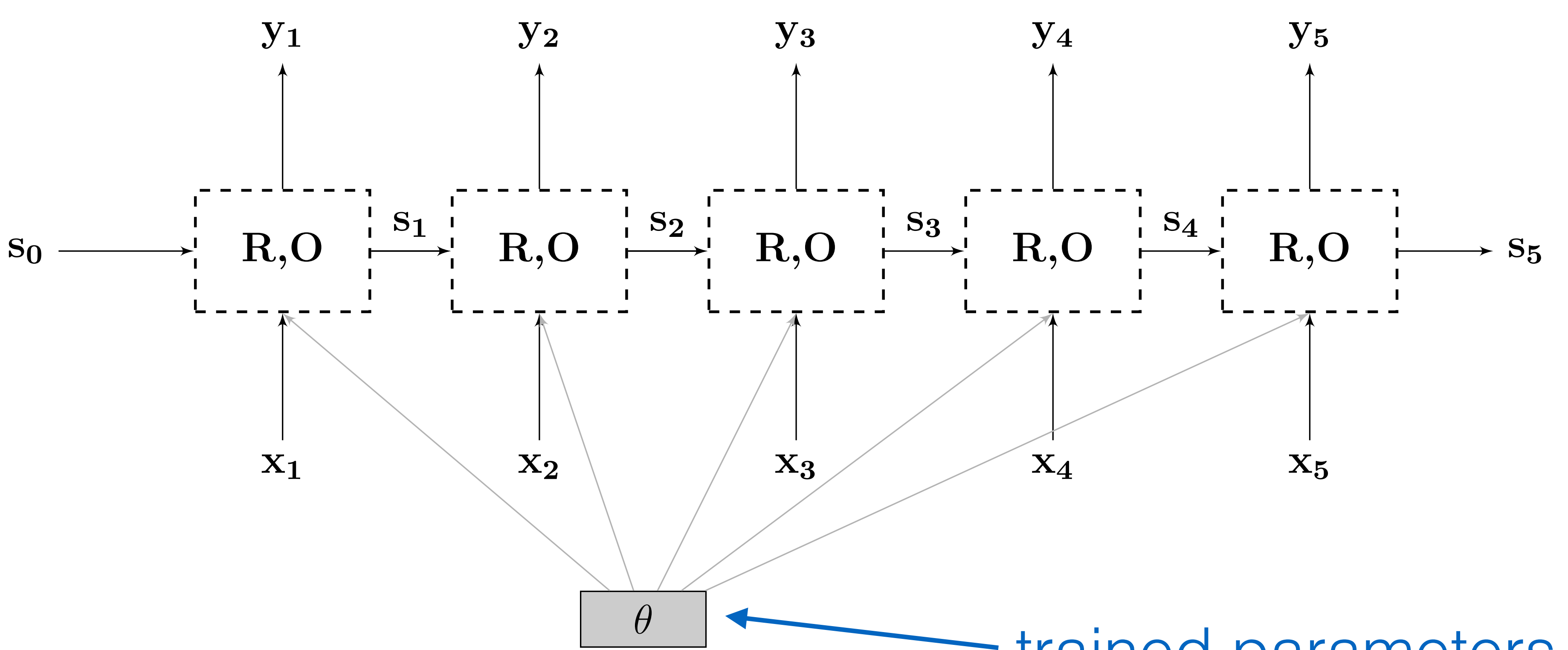
- On their own? **nothing**.
- **But we can train them.**

define function form

define loss

Recurrent Neural Networks

- What are the vectors $\mathbf{y_i}$ good for?



- On their own? **nothing**.

- **But we can train them.**

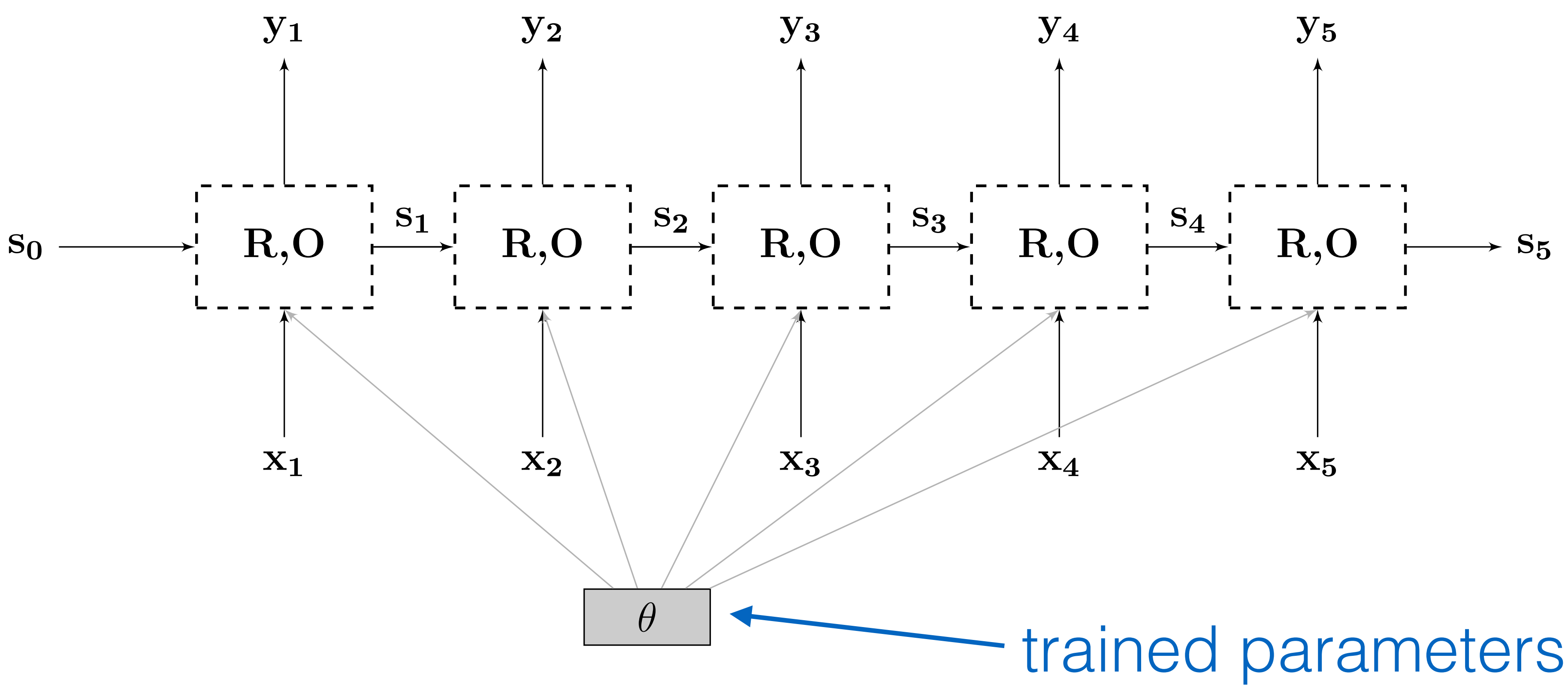
define function form

define loss

SimpleRNN:

$$R_{SRNN}(\mathbf{s_{i-1}}, \mathbf{x_i}) = tanh(\mathbf{W^s} \cdot \mathbf{s_{i-1}} + \mathbf{W^x} \cdot \mathbf{x_i})$$

looks simple.
theoretically powerful.
practically, not so much.



- On their own? **nothing**.
- **But we can train them.**

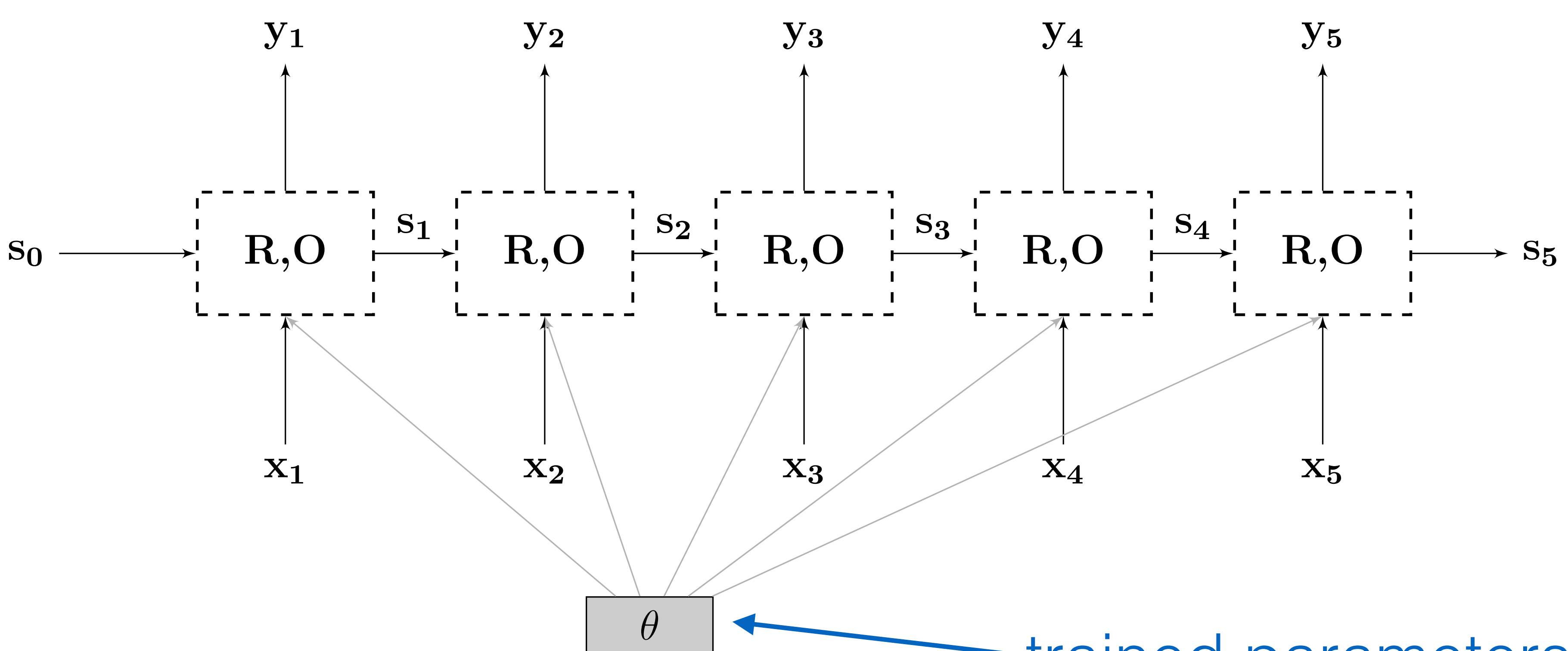
define function form

define loss

LSTM:

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

looks complex, and is.
very strong in practice.



trained parameters.

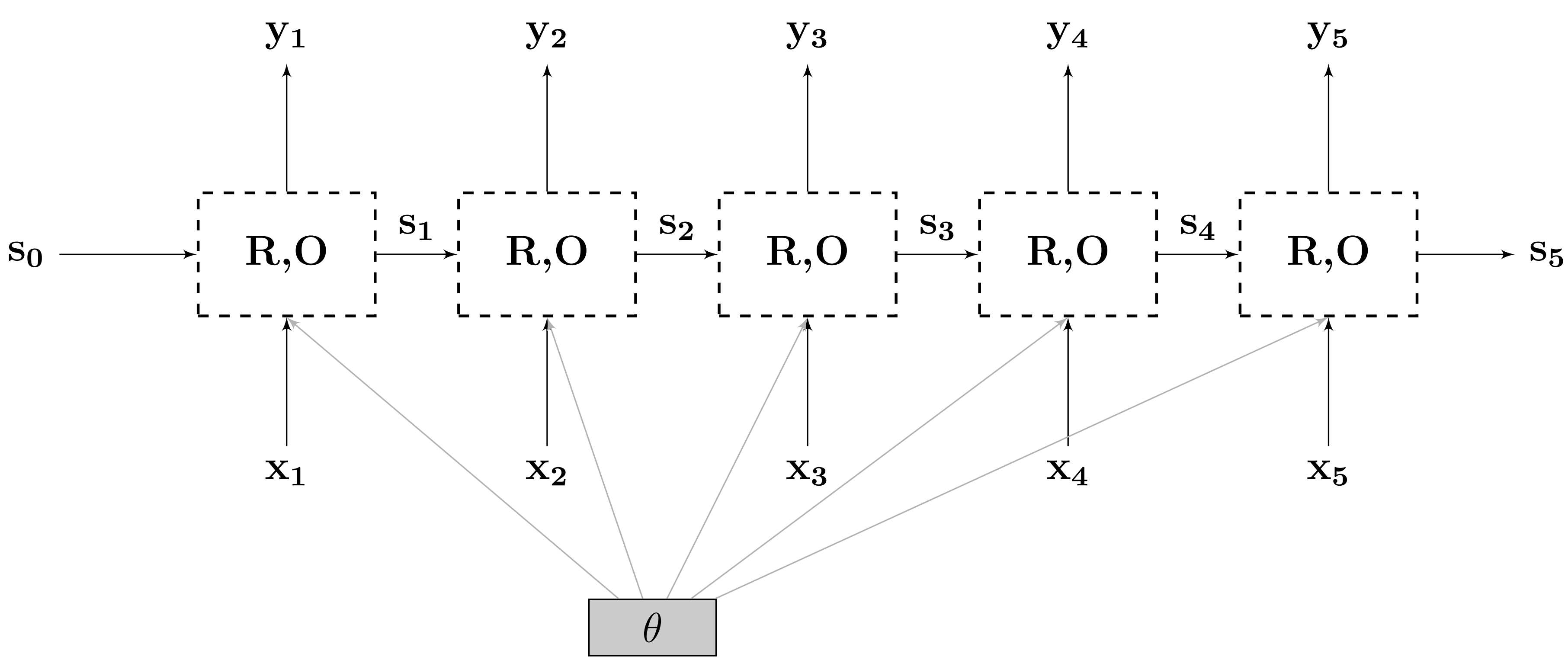
- On their own? **nothing**.
- **But we can train them.**

define function form

define loss

Recurrent Neural Networks

- What are the vectors $\mathbf{y_i}$ good for?



- On their own? **nothing**.

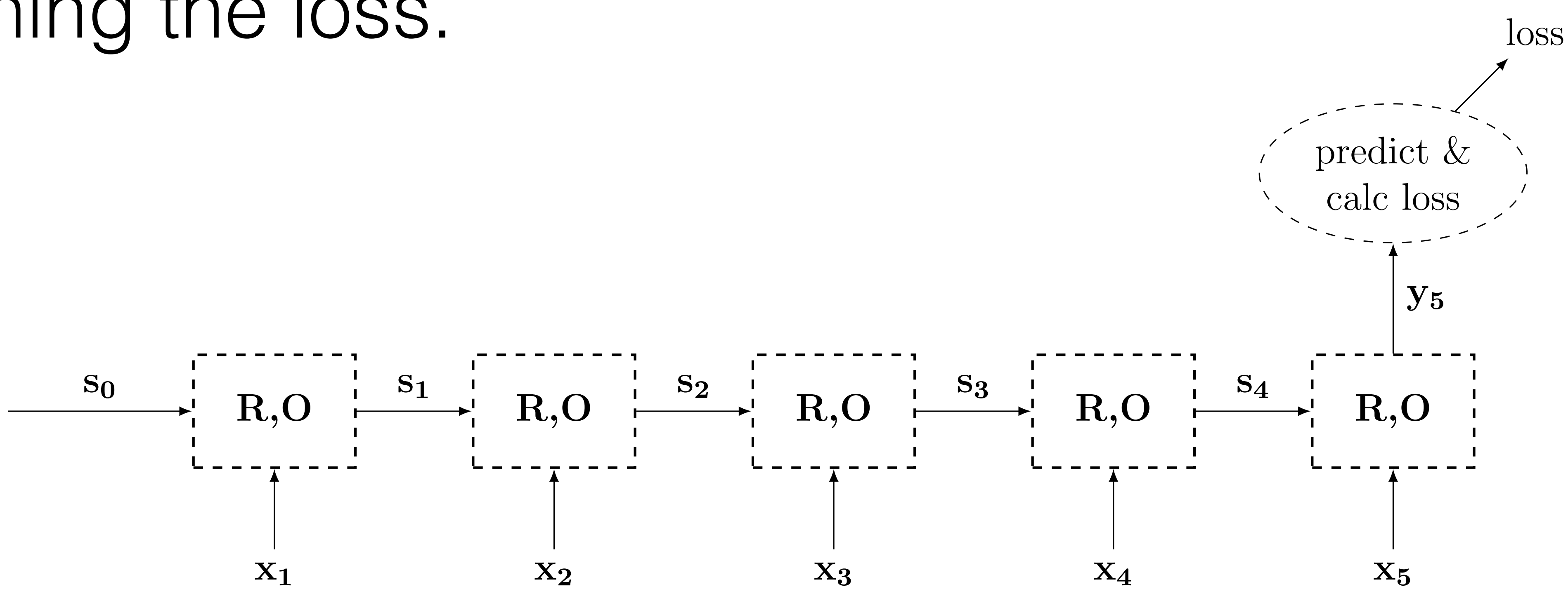
- **But we can train them.**

define function form

define loss

Recurrent Neural Networks

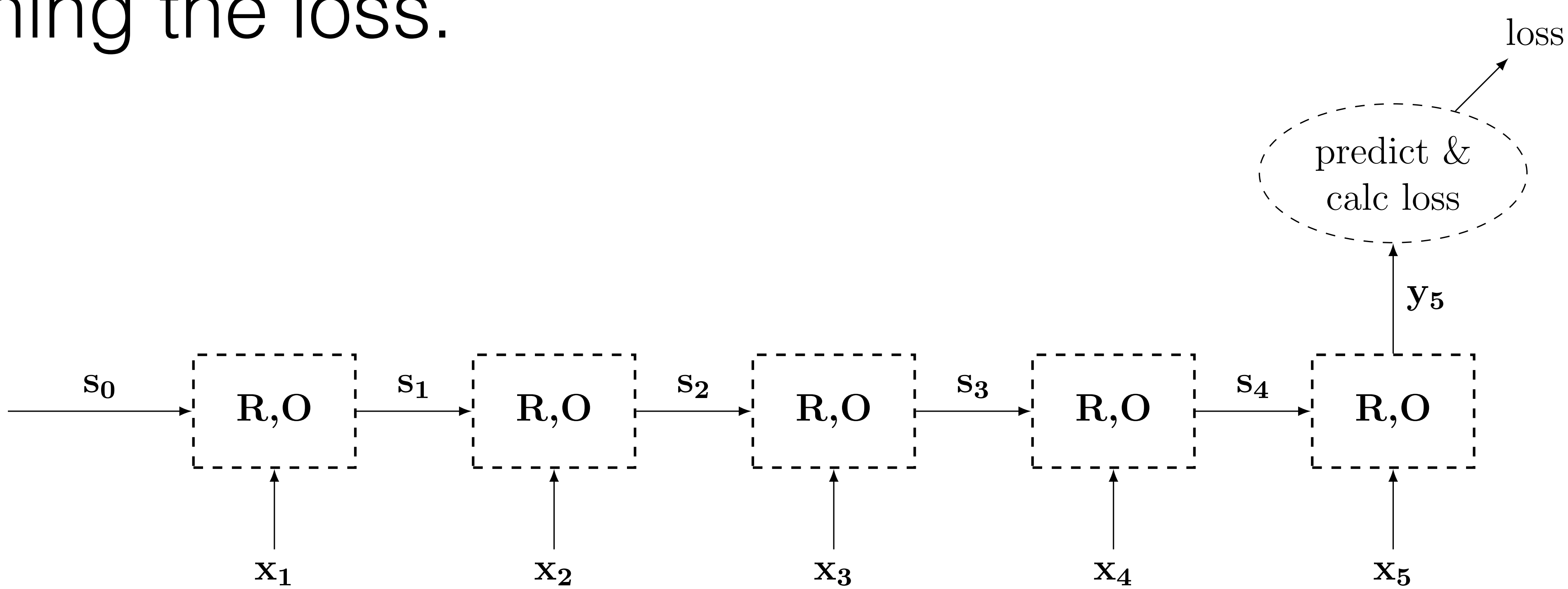
Defining the loss.



Acceptor: predict something from end state.
Backprop the error all the way back.
Train the network to capture meaningful information

Recurrent Neural Networks

Defining the loss.

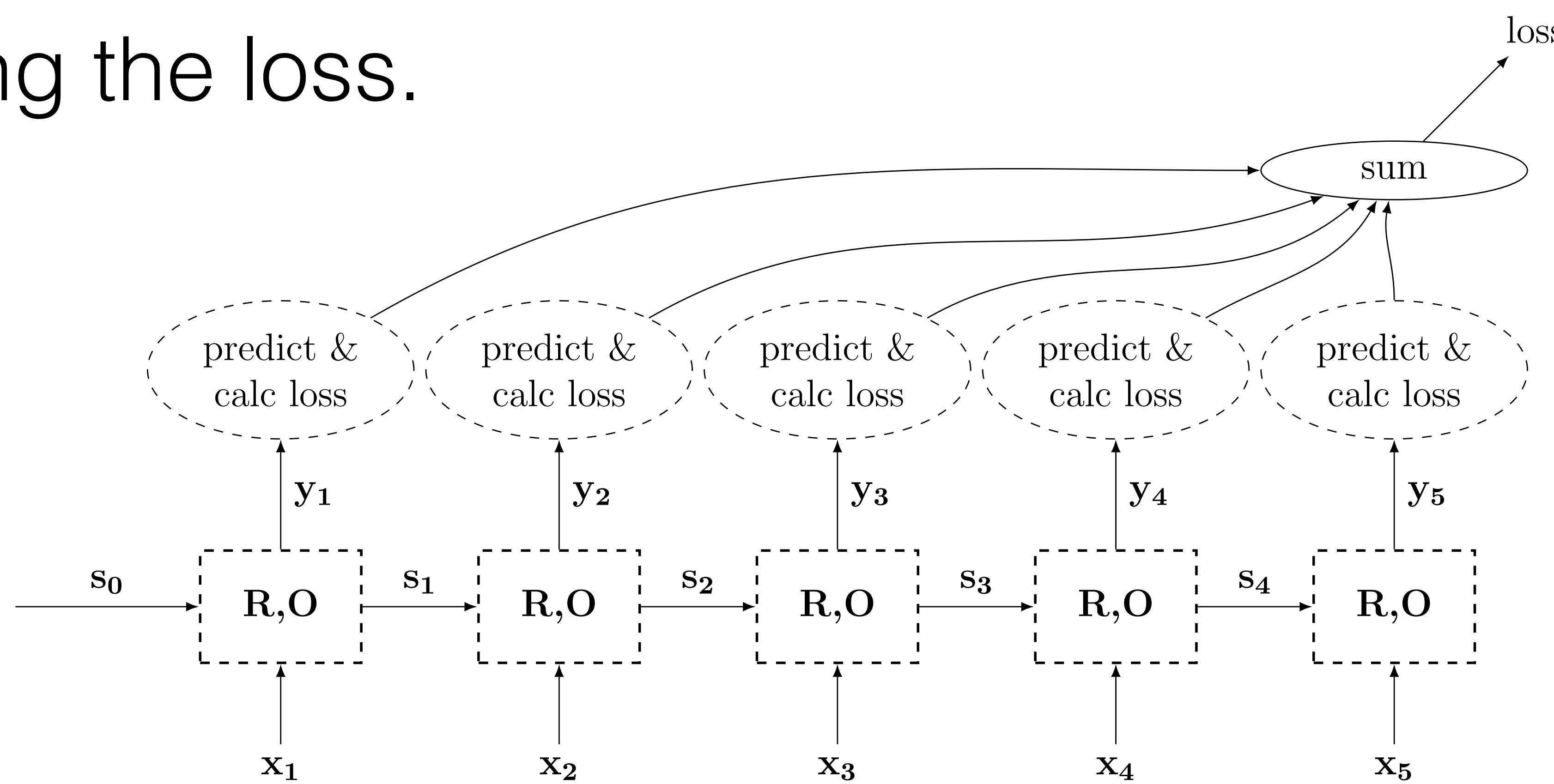


the final vector is a good "summary" of the sequence

Acceptor: predict something from end state.
Backprop the error all the way back.
Train the network to capture meaningful information

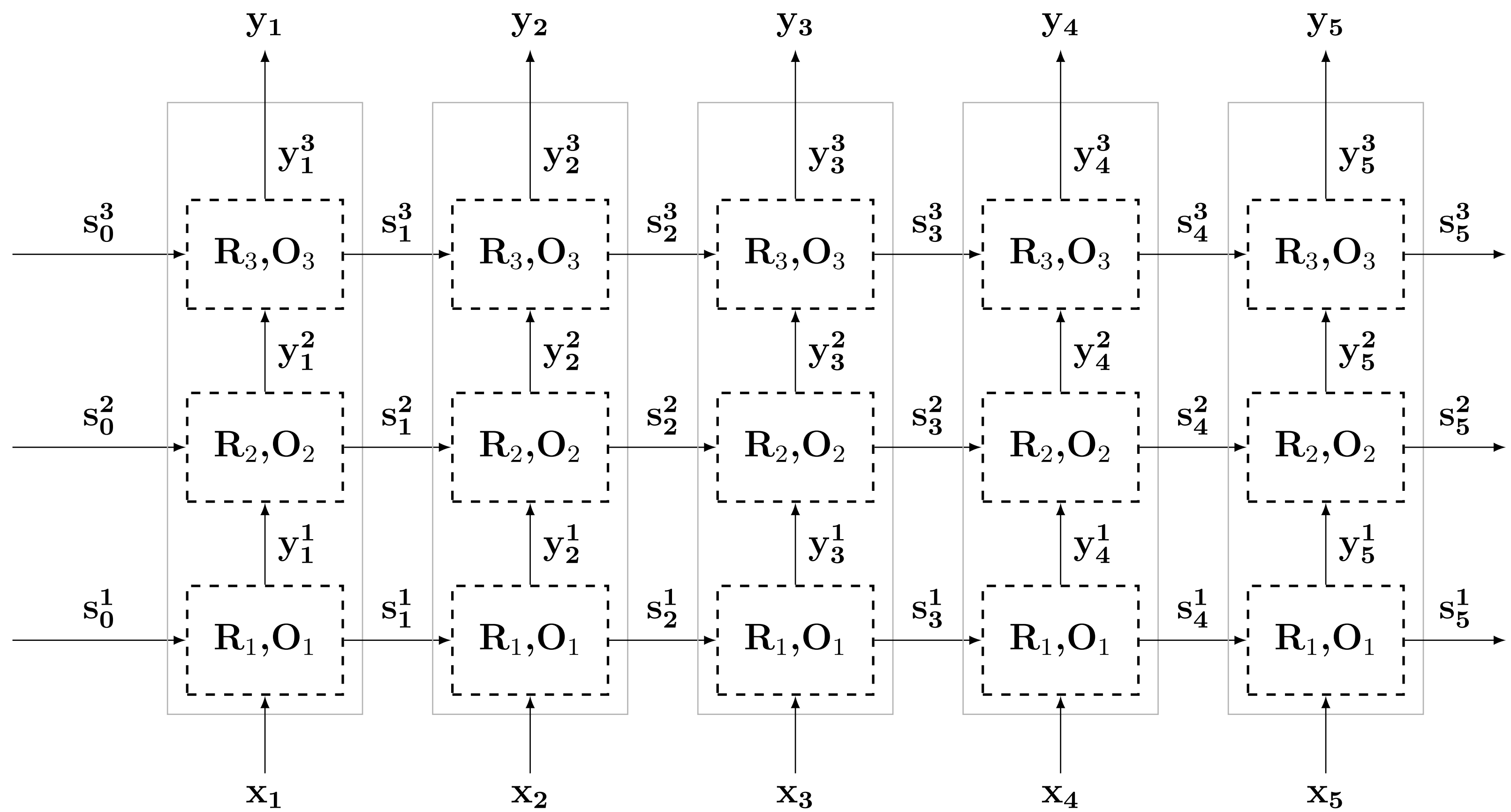
Recurrent Neural Networks

Defining the loss.



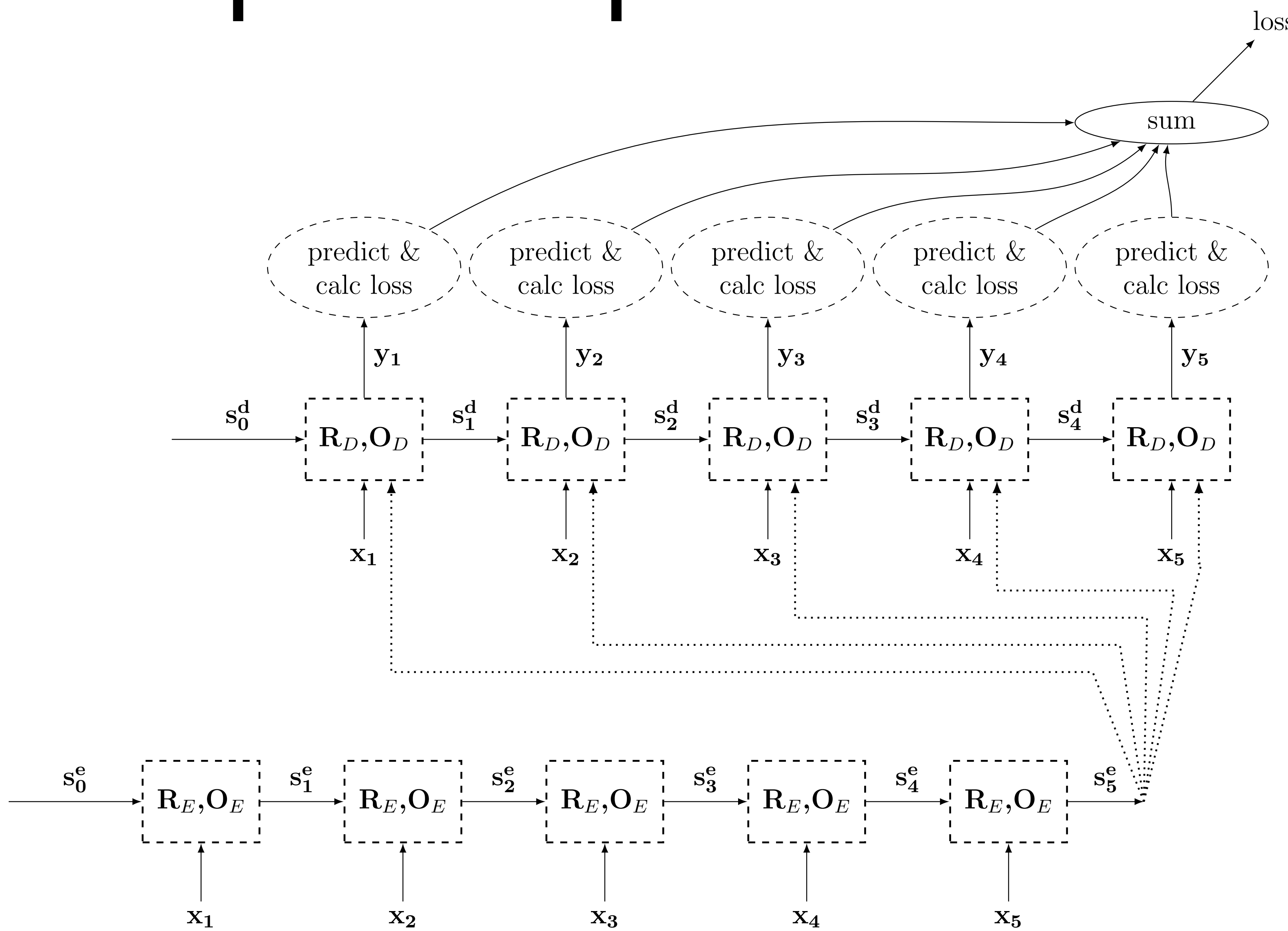
Transducer: predict something from each state.
Backprop the sum of errors all the way back.
Train the network to capture meaningful information

"Deep RNNs"



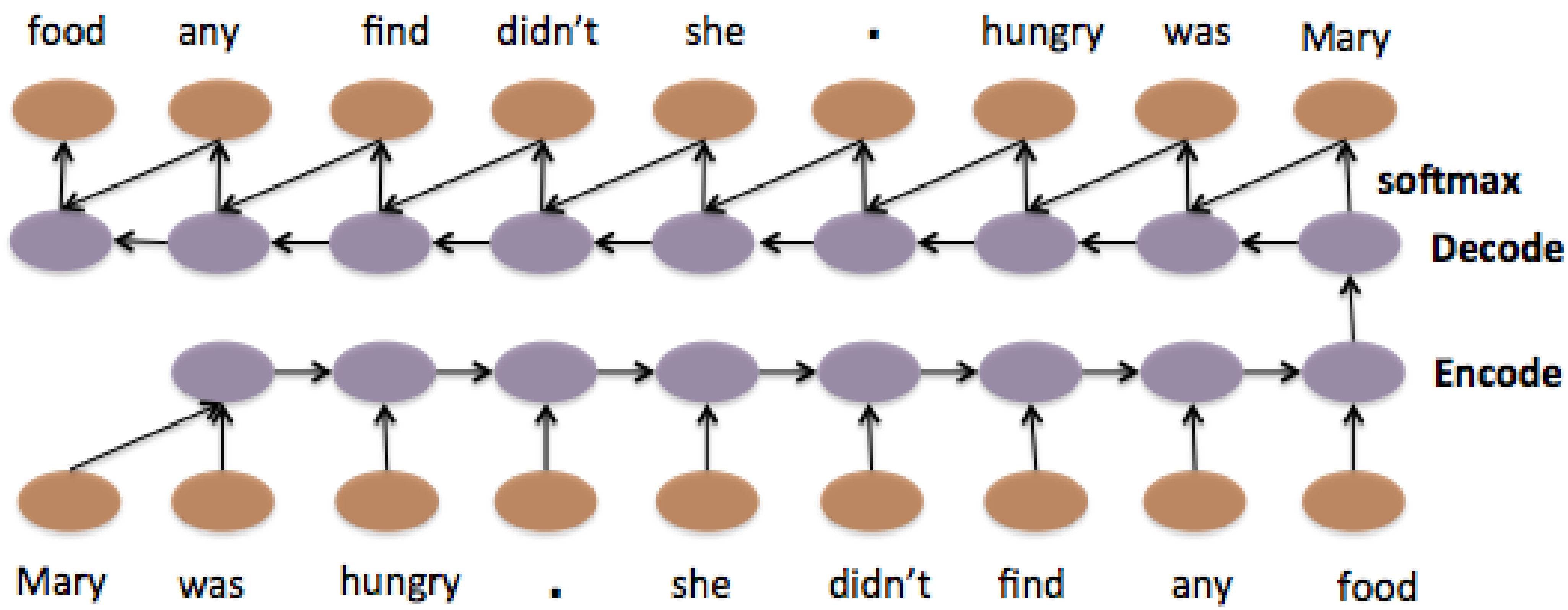
RNN can be stacked
deeper is better!
(better how?)

seq2seq models



Encoder-decoder (seq2seq):
Encoder-RNN encodes the sentence.
Decoder RNN transduces something back.

Auto-Encoder

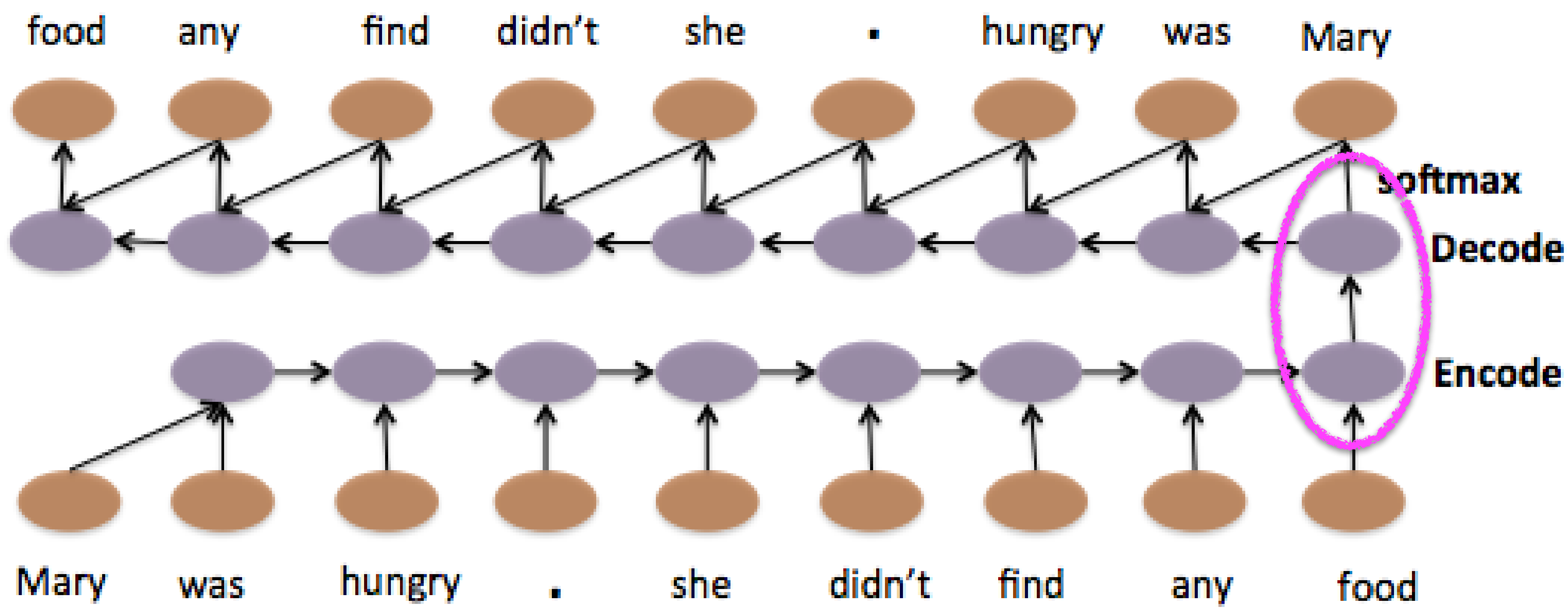


A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

Encoder-decoder (seq2seq):
Encoder encodes a sentence.
Decoder tries to reconstruct it.

Auto-Encoder

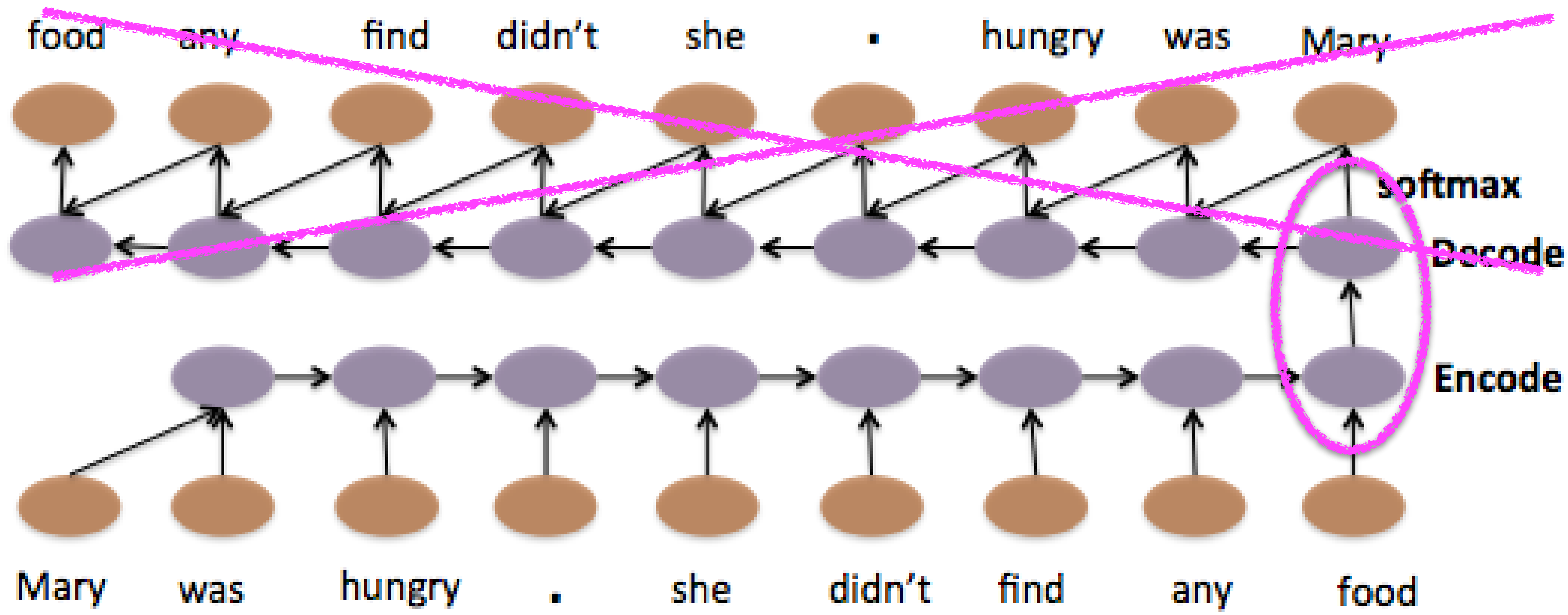


A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

encoded vector is a "generic sentence representation"

Auto-Encoder

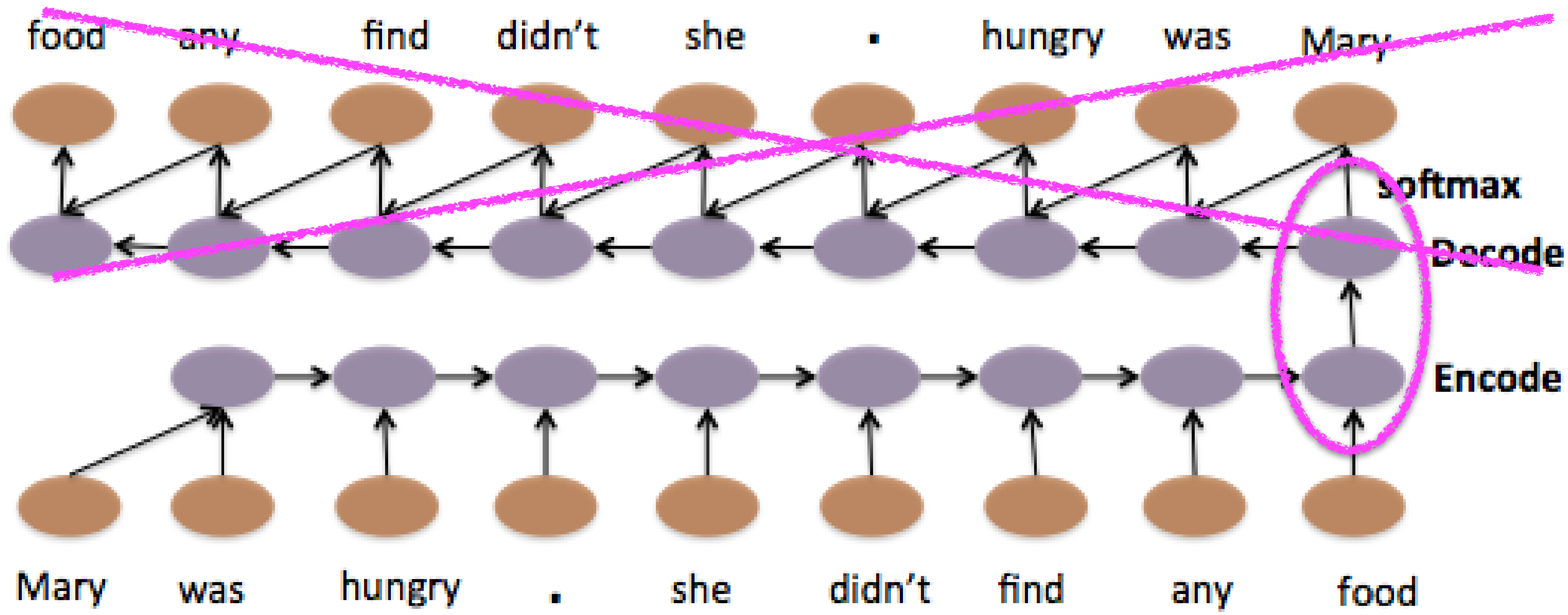


A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

encoded vector is a "generic sentence representation"

Auto-Encoder

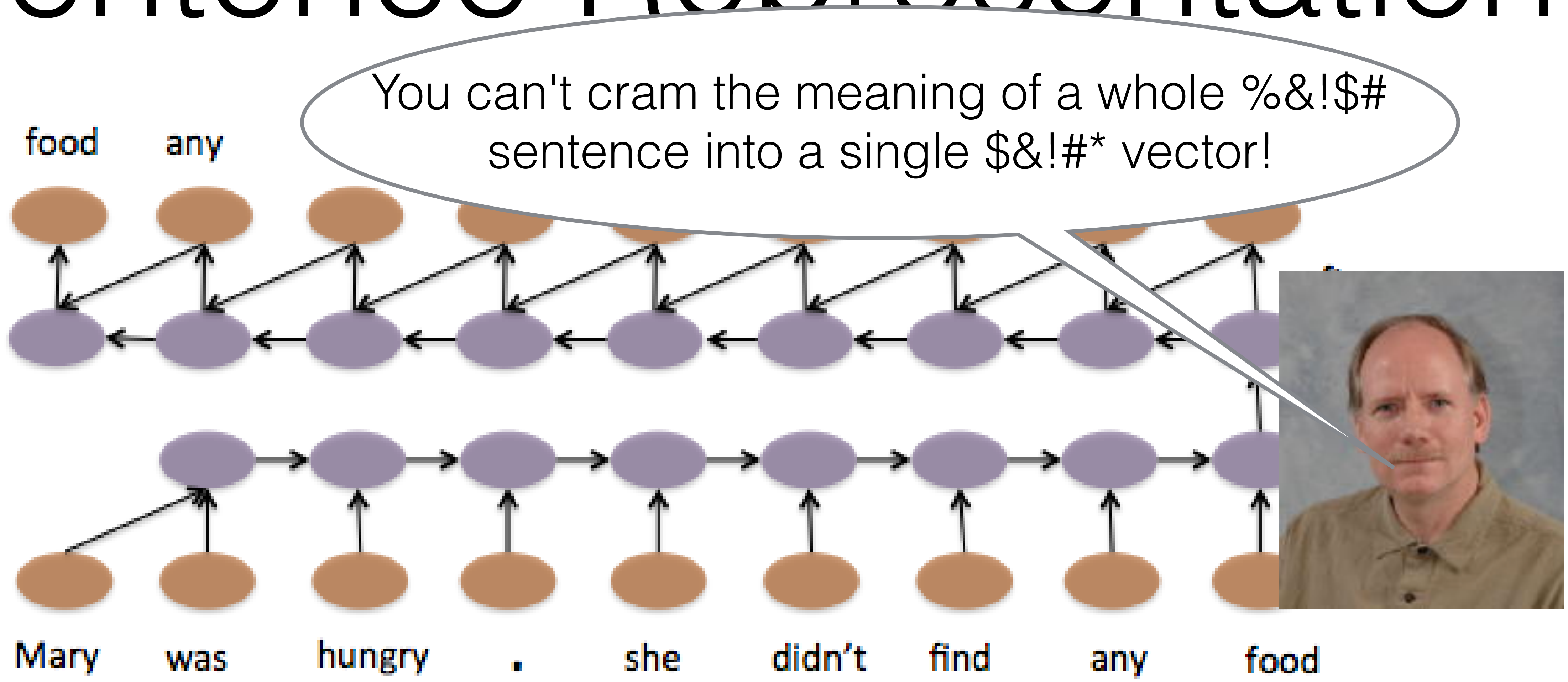


A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

encoded vector is a "generic sentence representation"

Sentence Representation

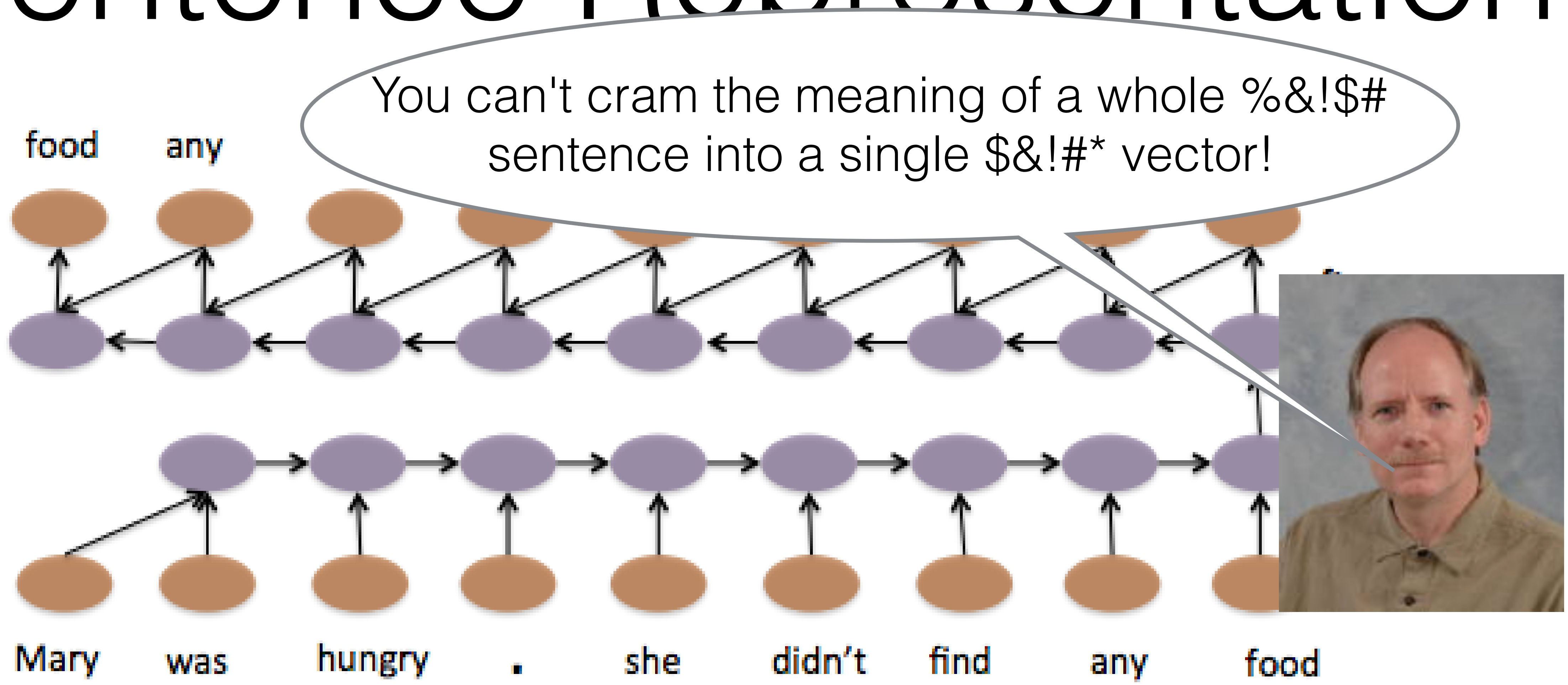


A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

encoded vector is a "generic sentence representation"

Sentence Representation



A Hierarchical Neural Autoencoder for Paragraphs and Documents

Jiwei Li, Minh-Thang Luong and Dan Jurafsky
Computer Science Department, Stanford University, Stanford, CA 94305, USA
jiweil, lmthang, jurafsky@stanford.edu

what **is** crammed into the encoded vector?

What is captured by the encoded vector?

BIU NLP

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]

IBM Research

Published as a conference paper at ICLR 2017

Rejected from pretty much all NLP venues

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.

Our Goal

Analyze and compare sentence representations in task and model independent manner

The Idea

- What information is encoded in the vector?
- **Let's ask it!**
- Design tasks to query specific kinds of information.
- Train a model to solve them, and see how well it does.
- A mechanism for comparing different sentence representations.

The Idea

- What information is encoded in the vector?
- **Let's ask it!**
- Design tasks to query specific kinds of information.
- Train a model to solve them, and see how well it does.
- A mechanism f
representations

If we can't train a classifier to act on information from a vector is the information really there?

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)

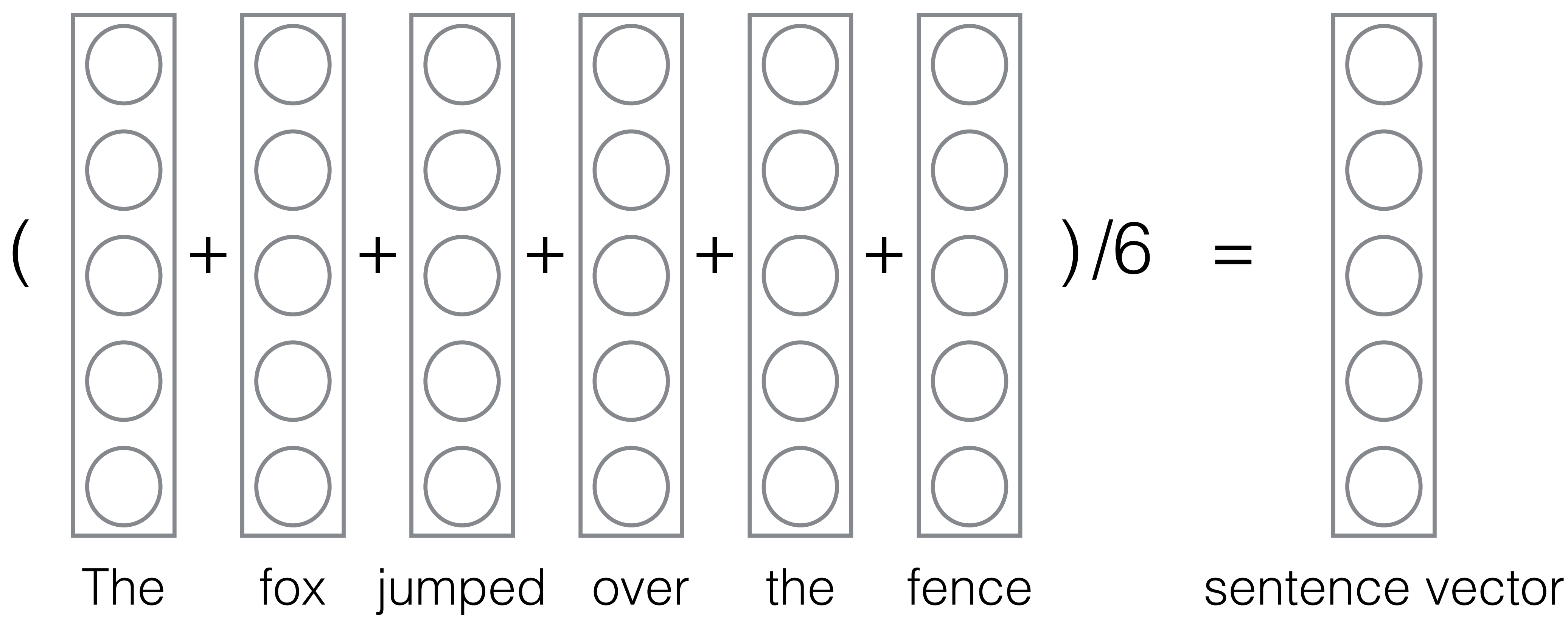
dim	acc
100	50%
300	80%
500	82%
750	79%
1000	83%

CBOW

??

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) dim	Encoder (LSTM) acc	CBOW
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

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.

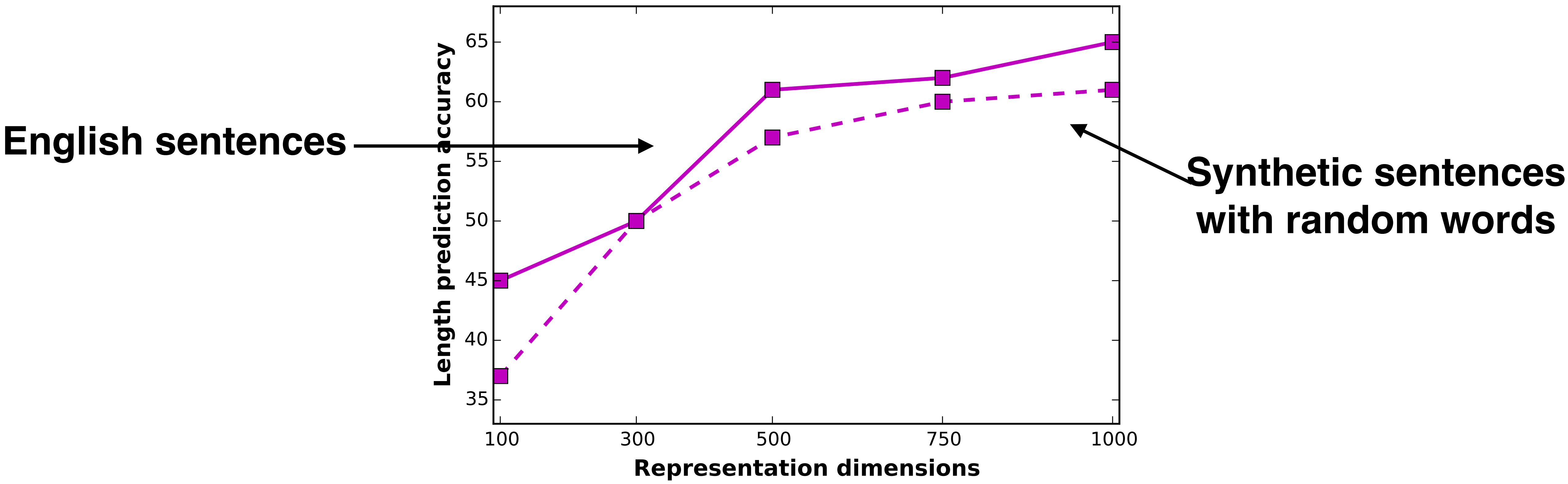
considering that CBOW is an averaged representation

How does CBOW encode length?

- Maybe some words are predictive of longer sentences?

How does CBOW encode length?

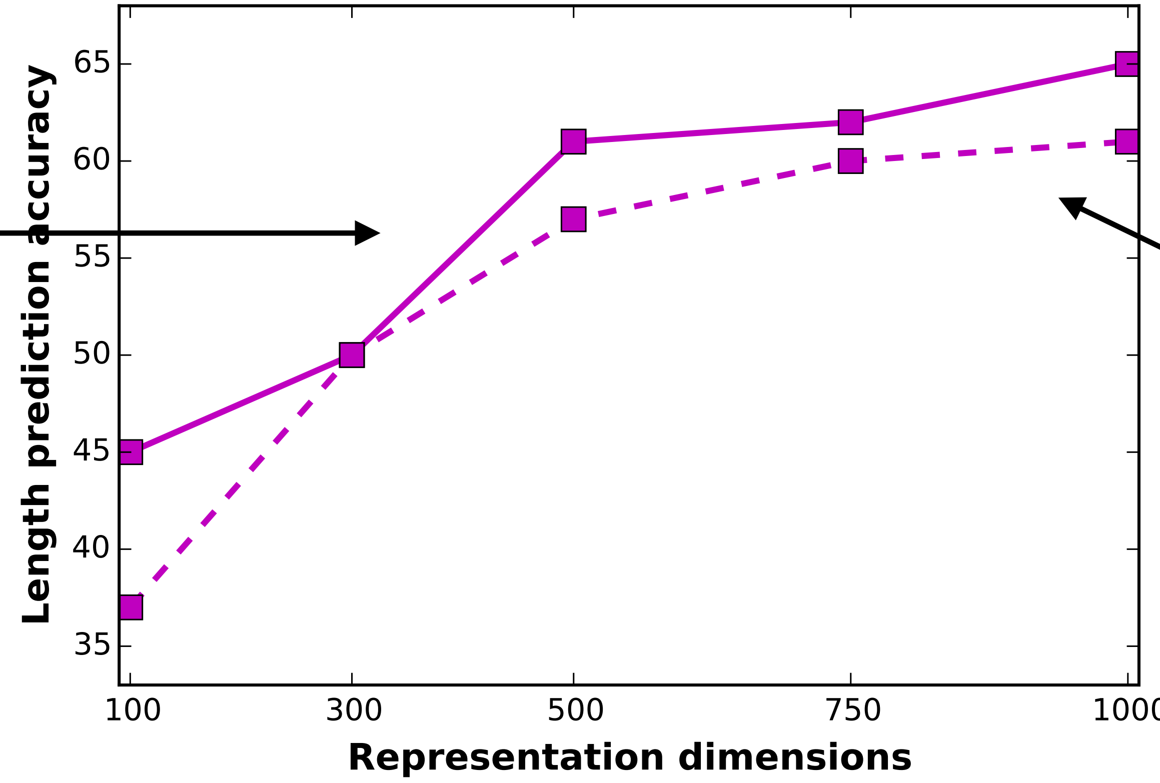
- Maybe some words are predictive of longer sentences?



How does CBOW encode length?

- Maybe some words are predictive of longer sentences?

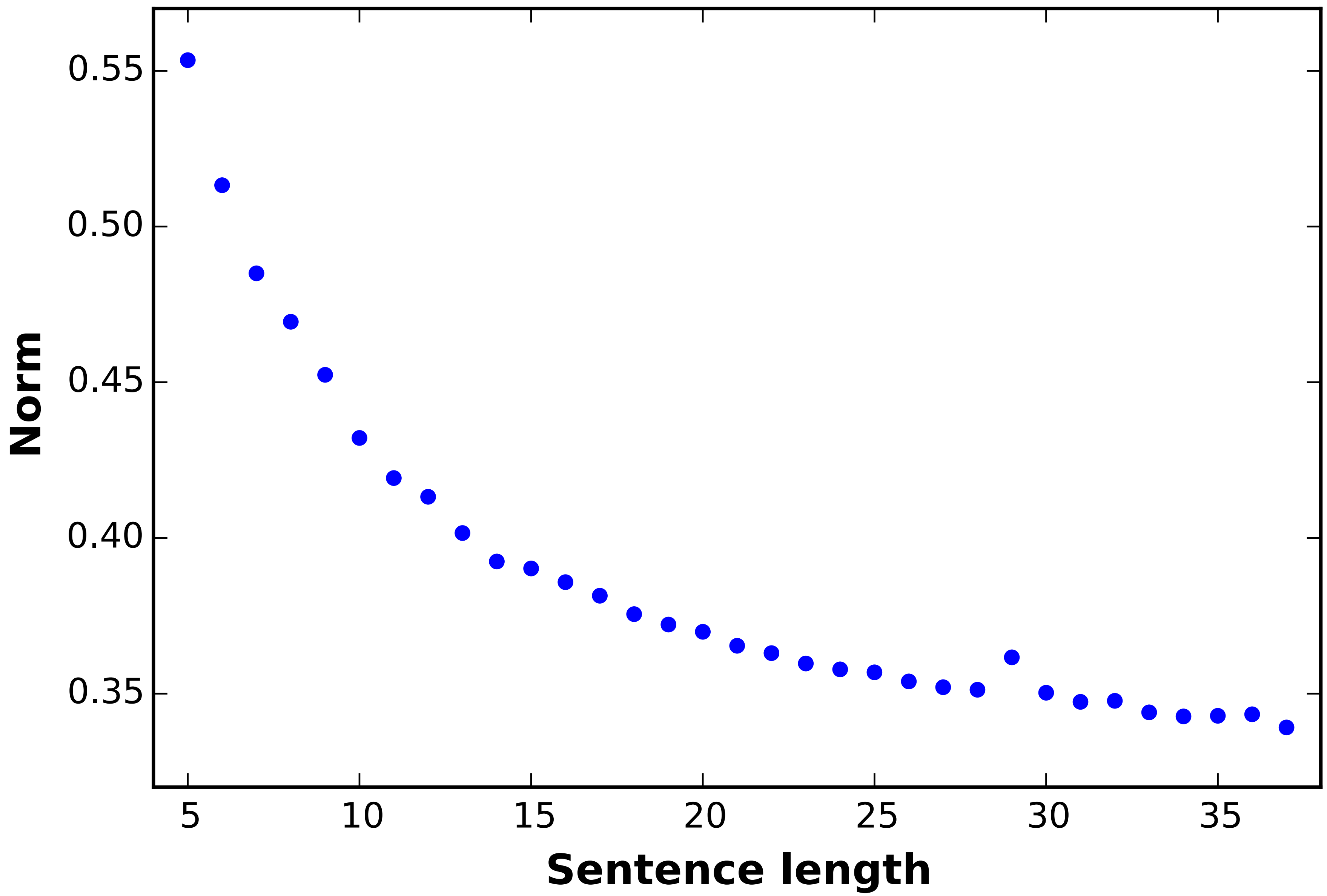
English sentences



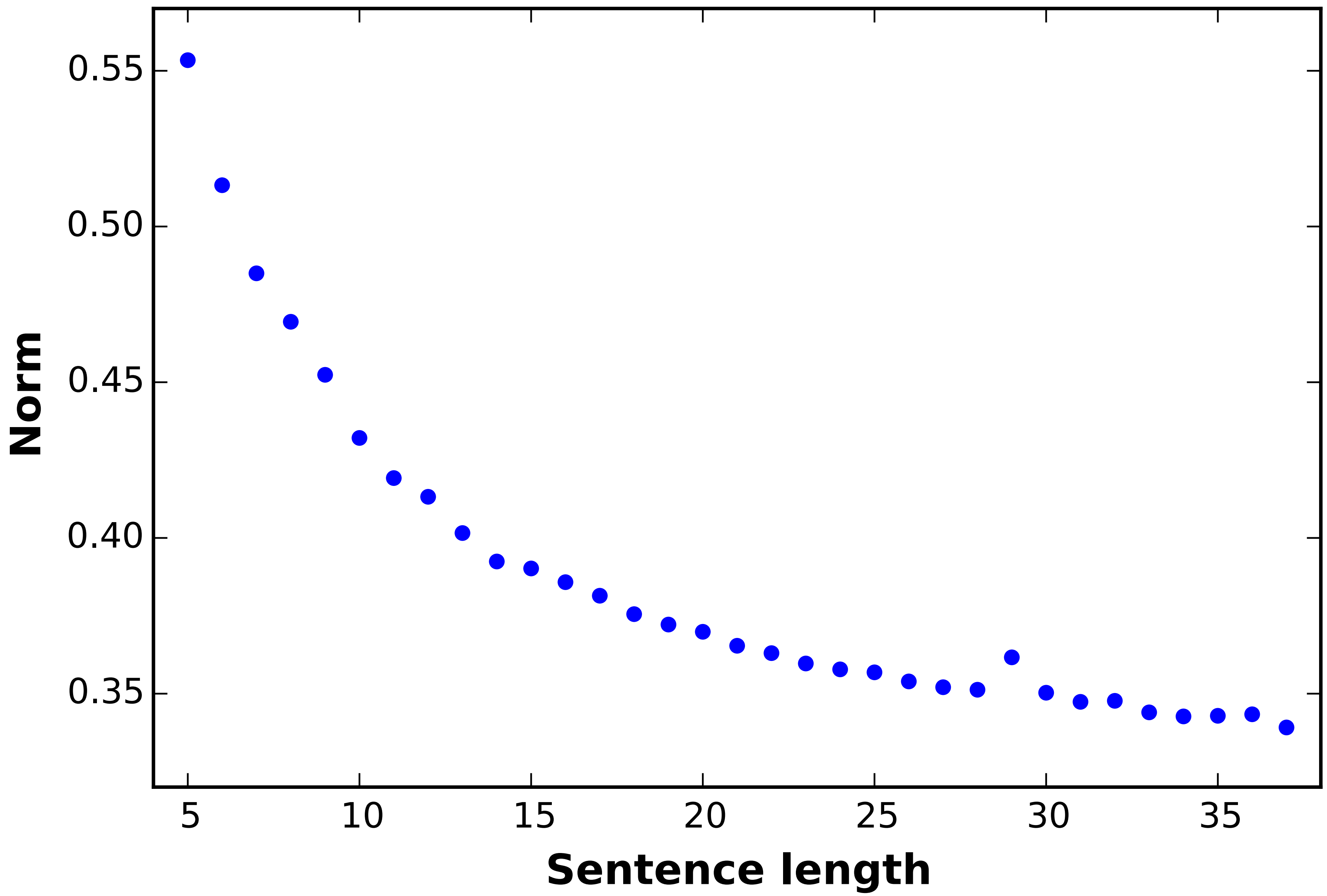
Synthetic sentences with random words

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%	

B I U
N L P

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) dim	Encoder (LSTM) acc	CBOW
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) dim	Encoder (LSTM) acc	CBOW
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

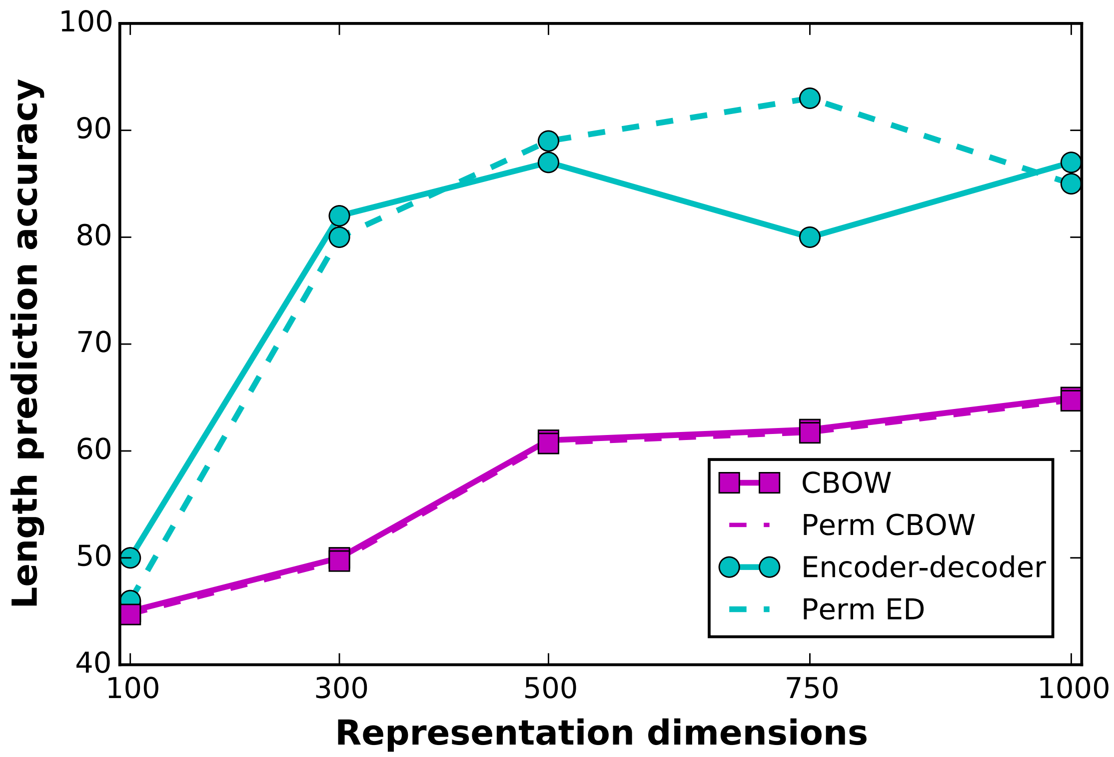
or Just Sequences?

- 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?

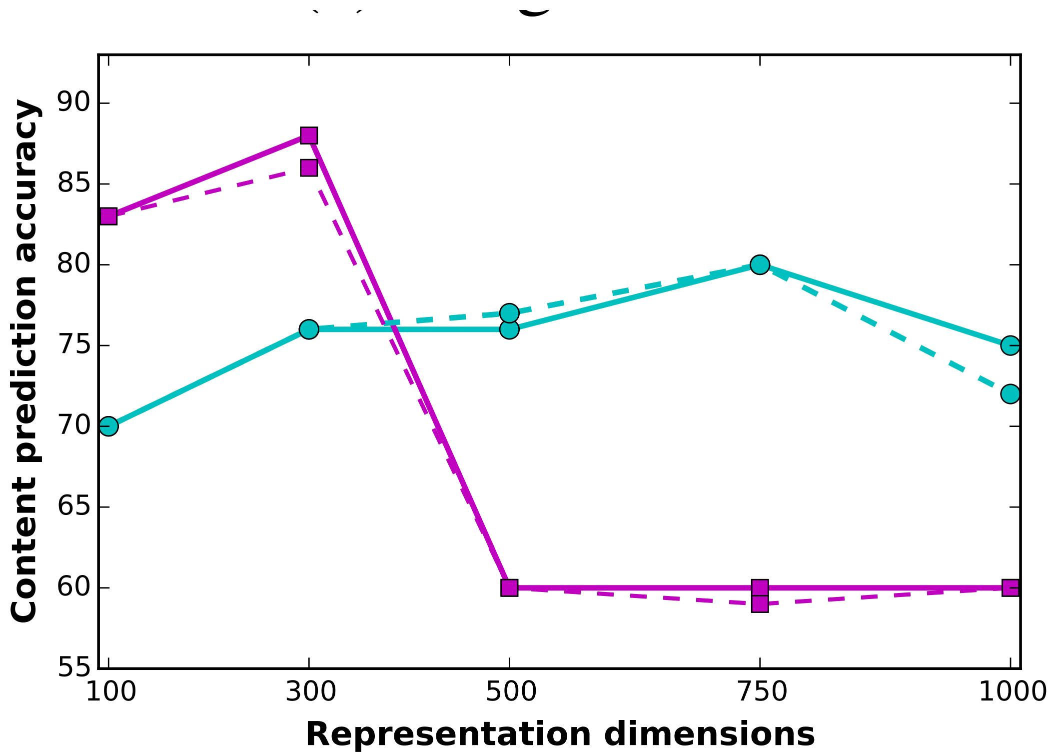


LSTM

cbow

Length Prediction

Does it Learn to Represent English or Just Sequences?

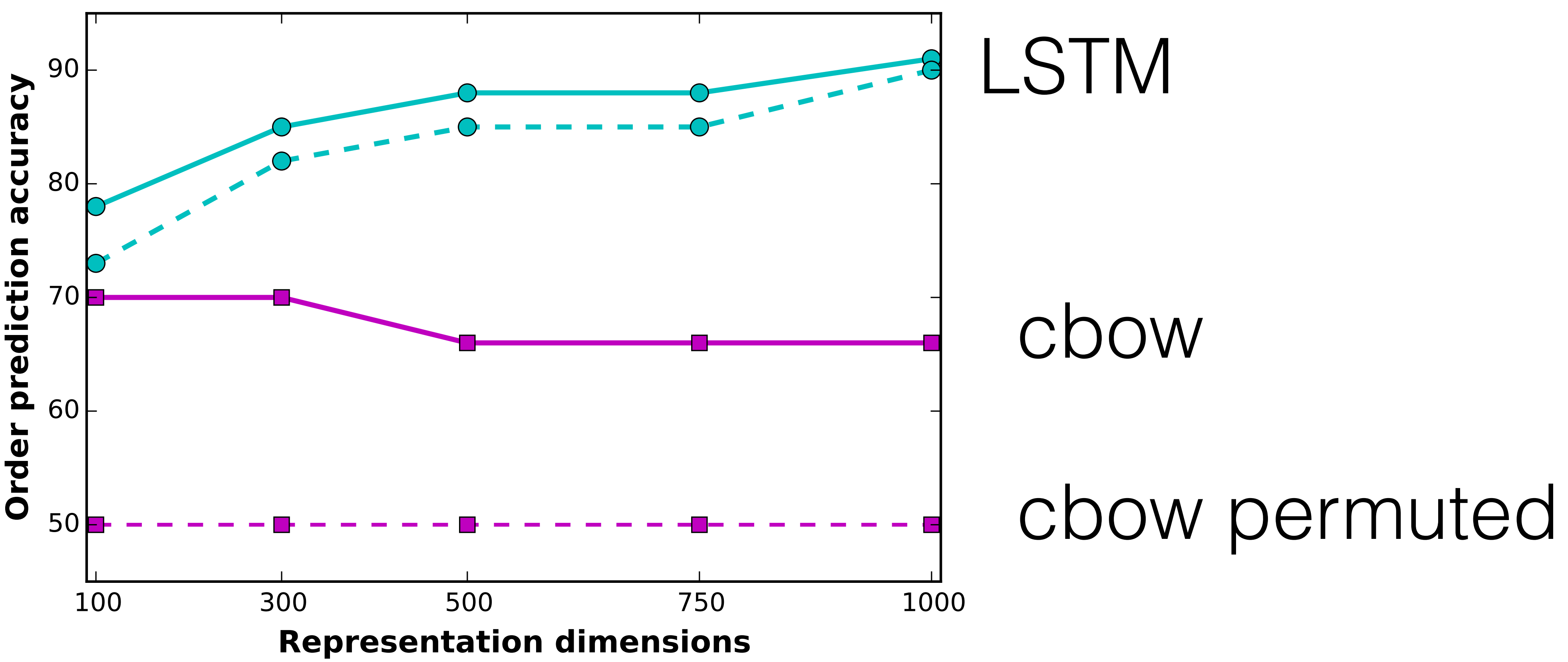


LSTM

cbow

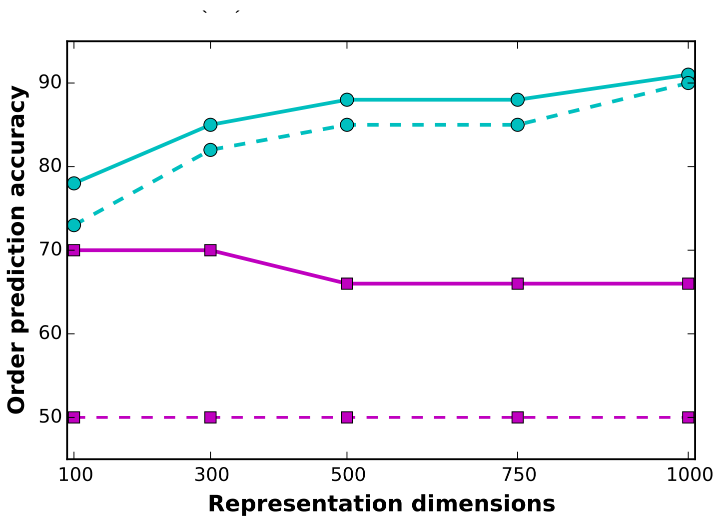
Content Prediction

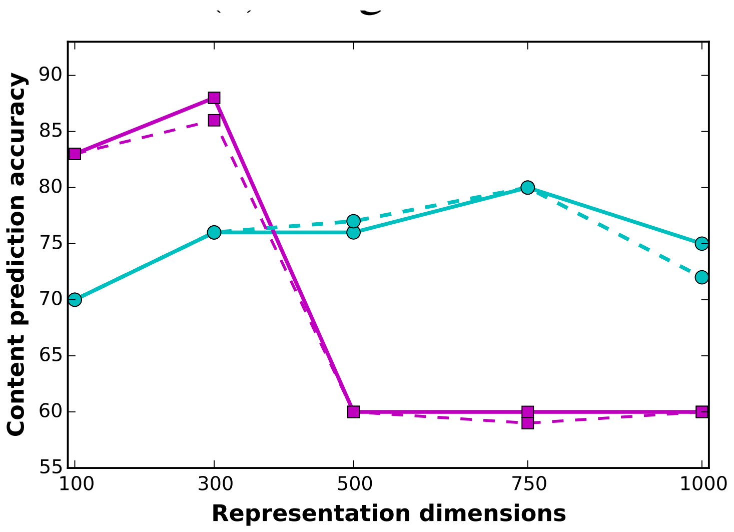
Does it Learn to Represent English or Just Sequences?

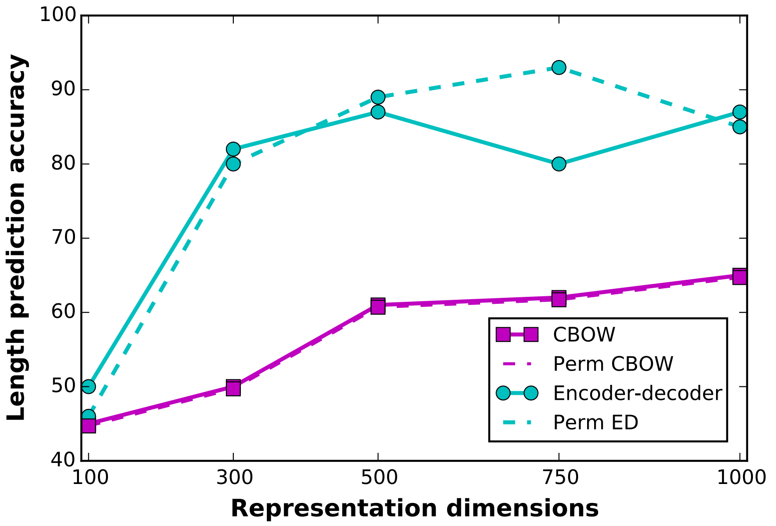


Order Prediction

Does it Learn to Represent English or Just Sequences?

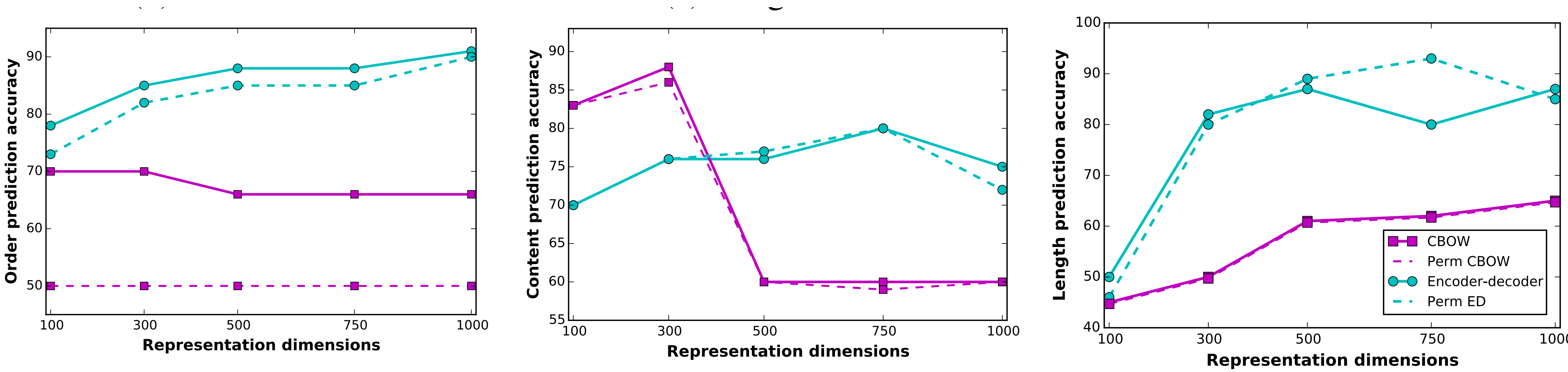






auto-encoder LSTM
does not really care what it encodes.
a generic sequence encoder.

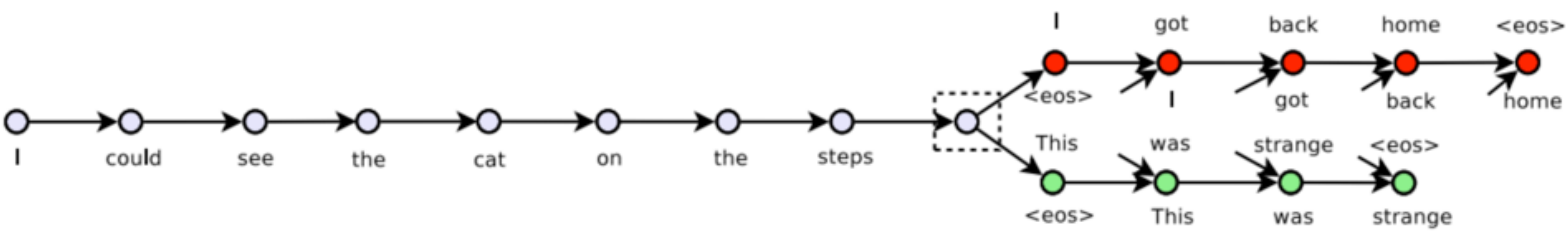
Does it Learn to Represent English or Just Sequences?



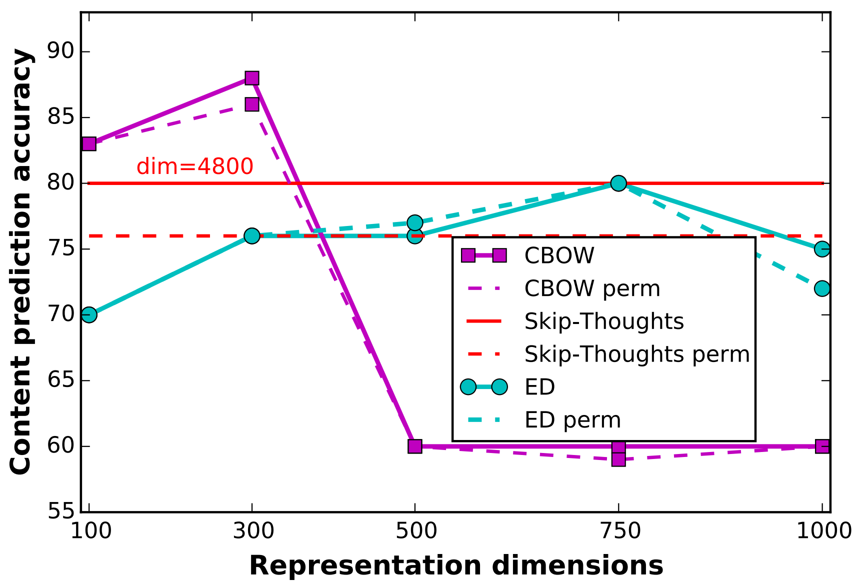
auto-encoder LSTM
does not really care what it encodes.
a generic sequence encoder.

nat-lang information is in the decoder.

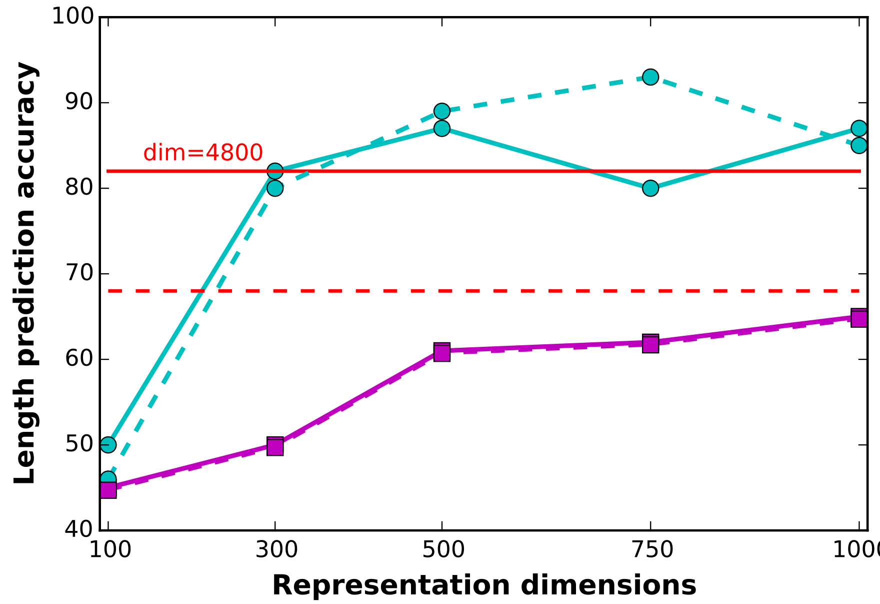
Skip-Thought Vectors



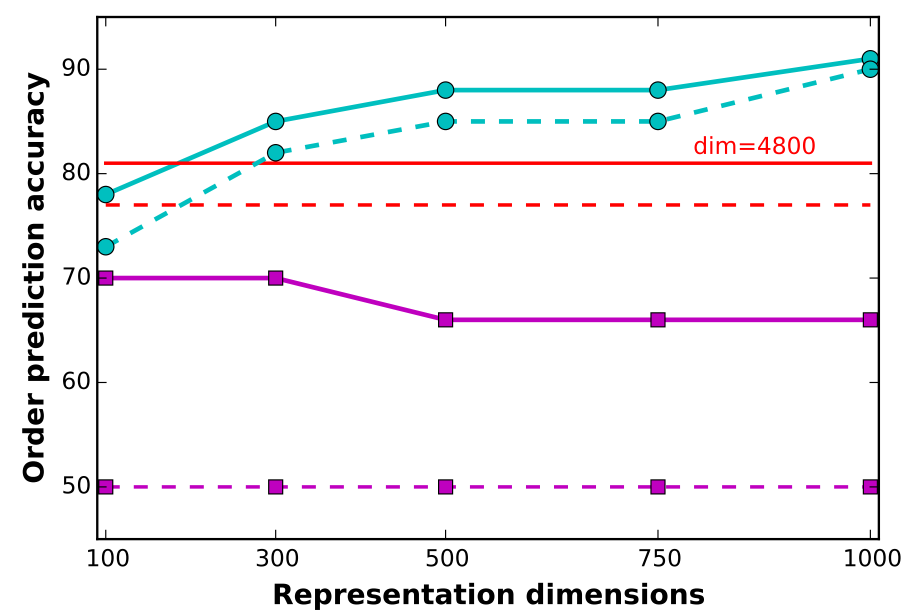
Does it Learn to Represent English or Just Sequences?



Content

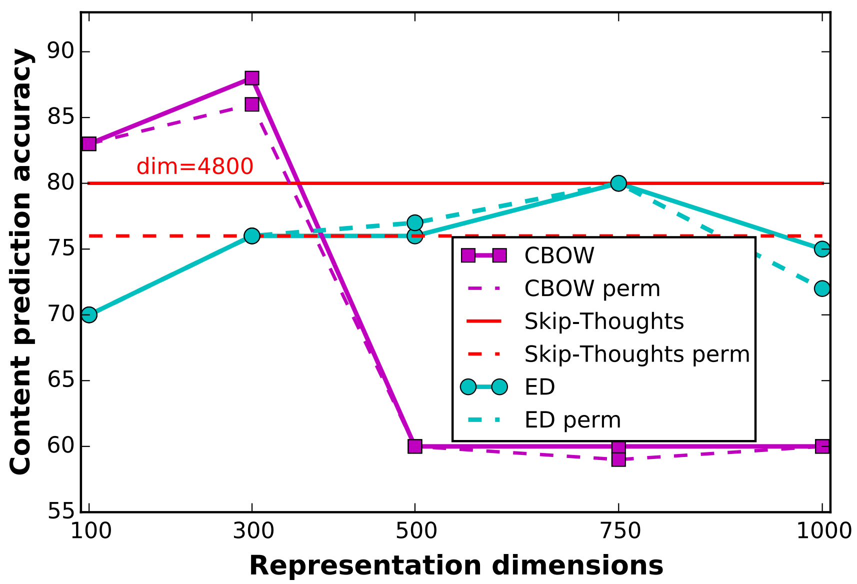


Length

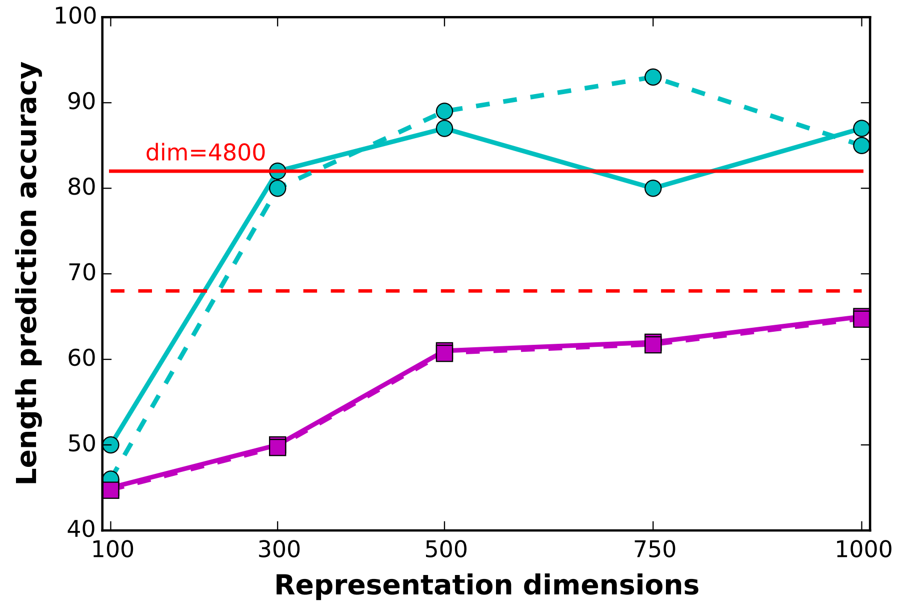


Order

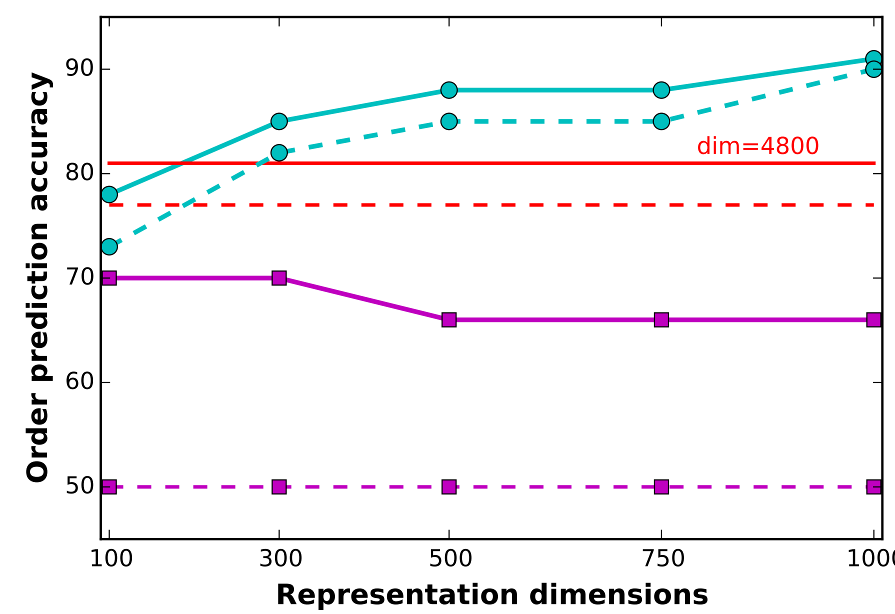
Does it Learn to Represent English or Just Sequences?



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.

RNNs and Hierarchical Structures

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

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)

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?

Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

predicts number with **99**% accuracy.

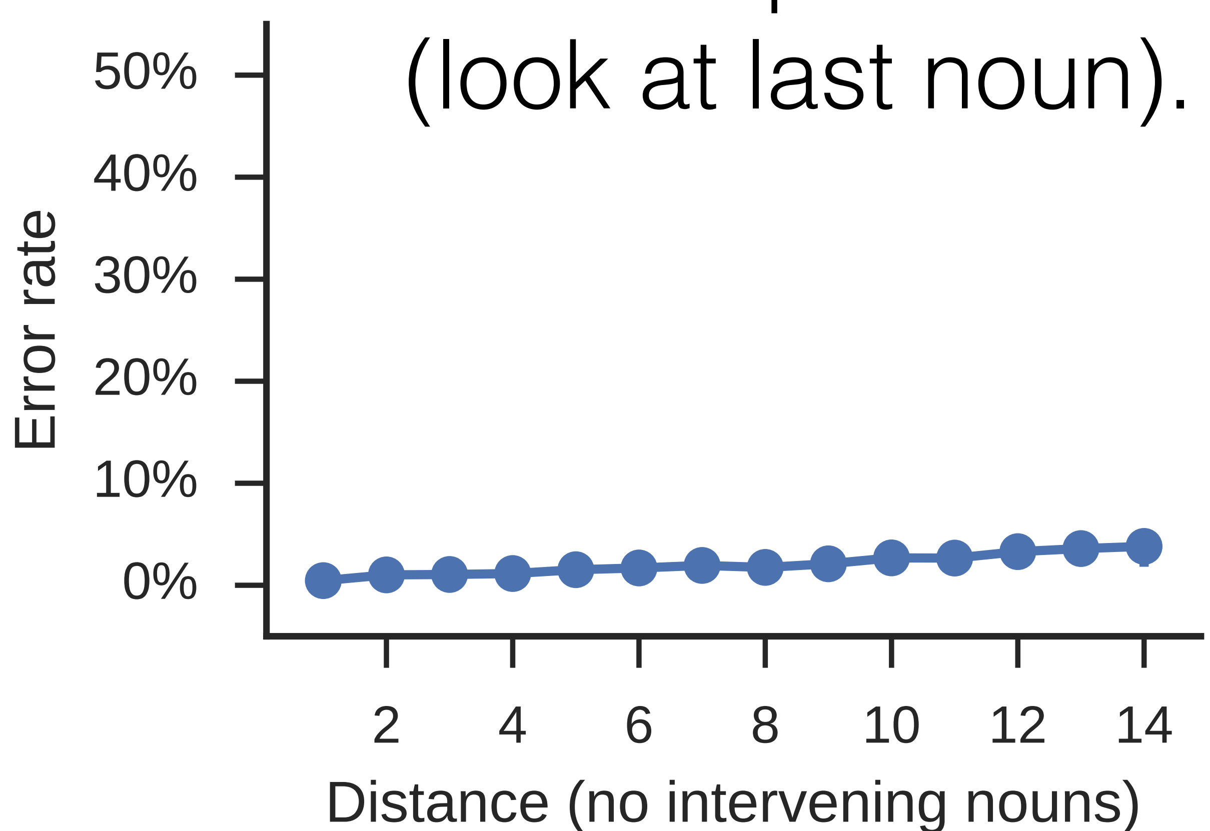
...but most examples are very easy
(look at last noun).

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?

$\boxed{\text{LSTMs learn agreement remarkably well.}}$

predicts number with **99**% accuracy.

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

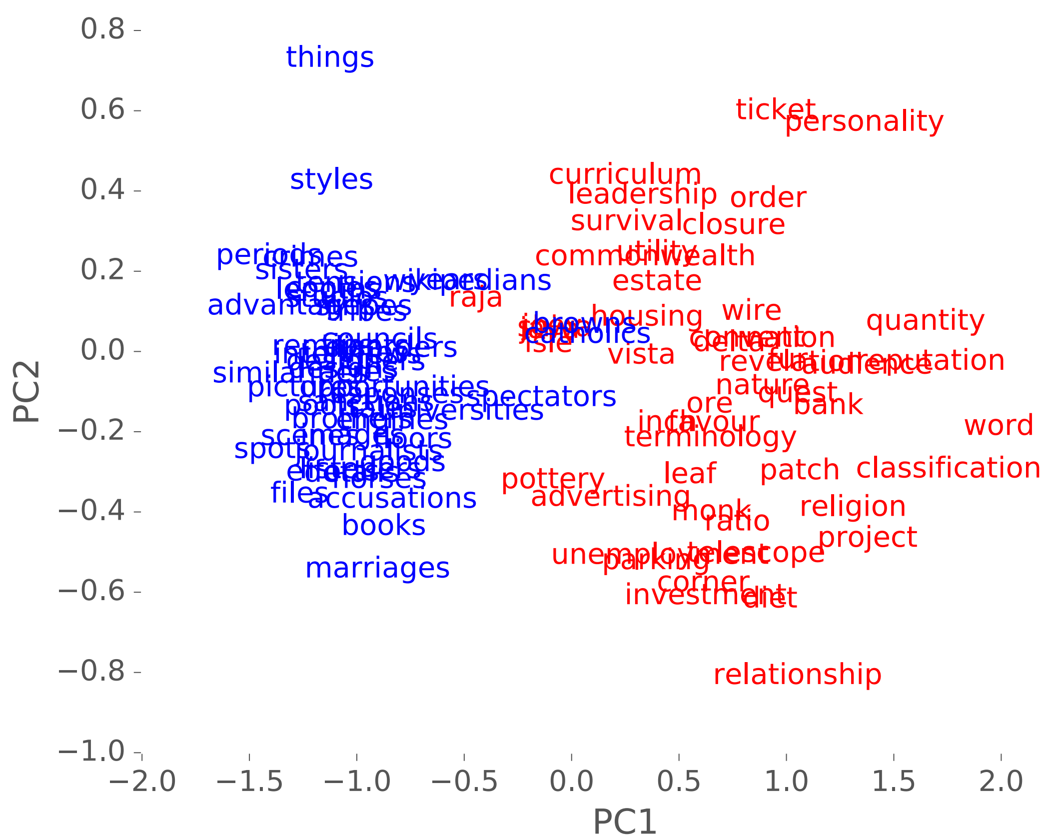
when restricted to cases
of at least one intervening noun:

97% accuracy

Can a sequence LSTM learn agreement?

LSTMs learn agreement remarkably well.

learns number of nouns

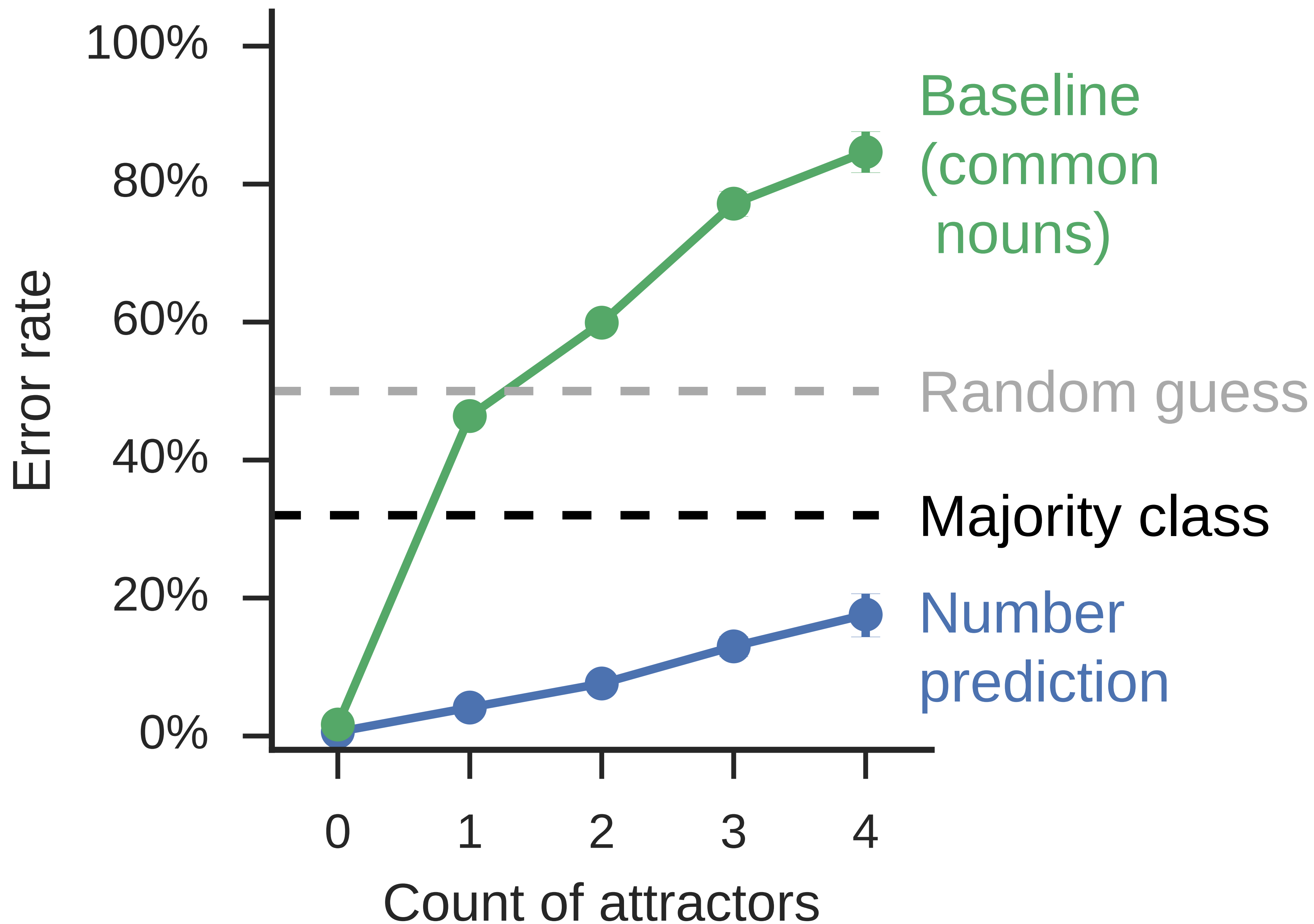


Can a sequence LSTM learn agreement?

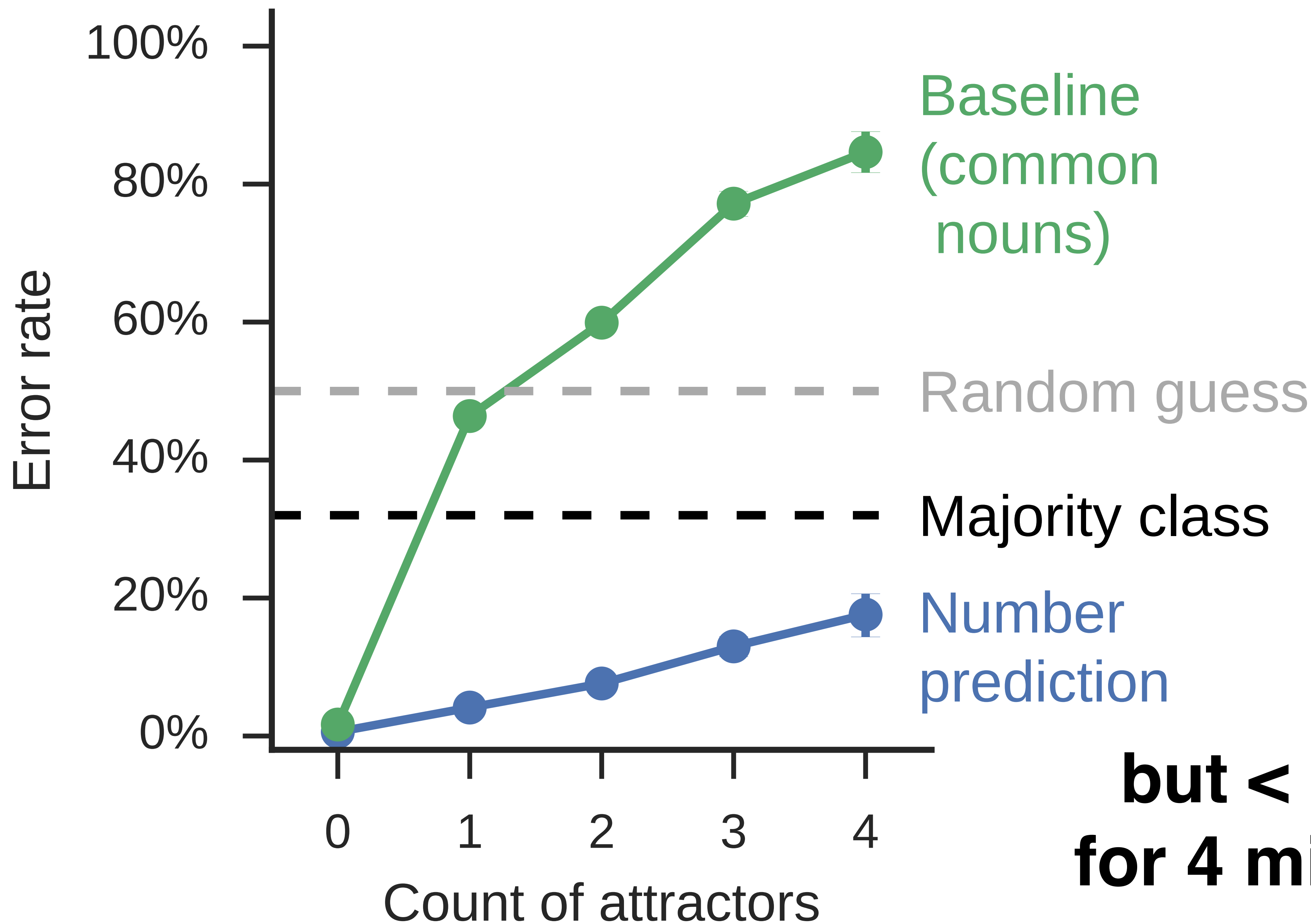
LSTMs learn agreement remarkably well.

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?

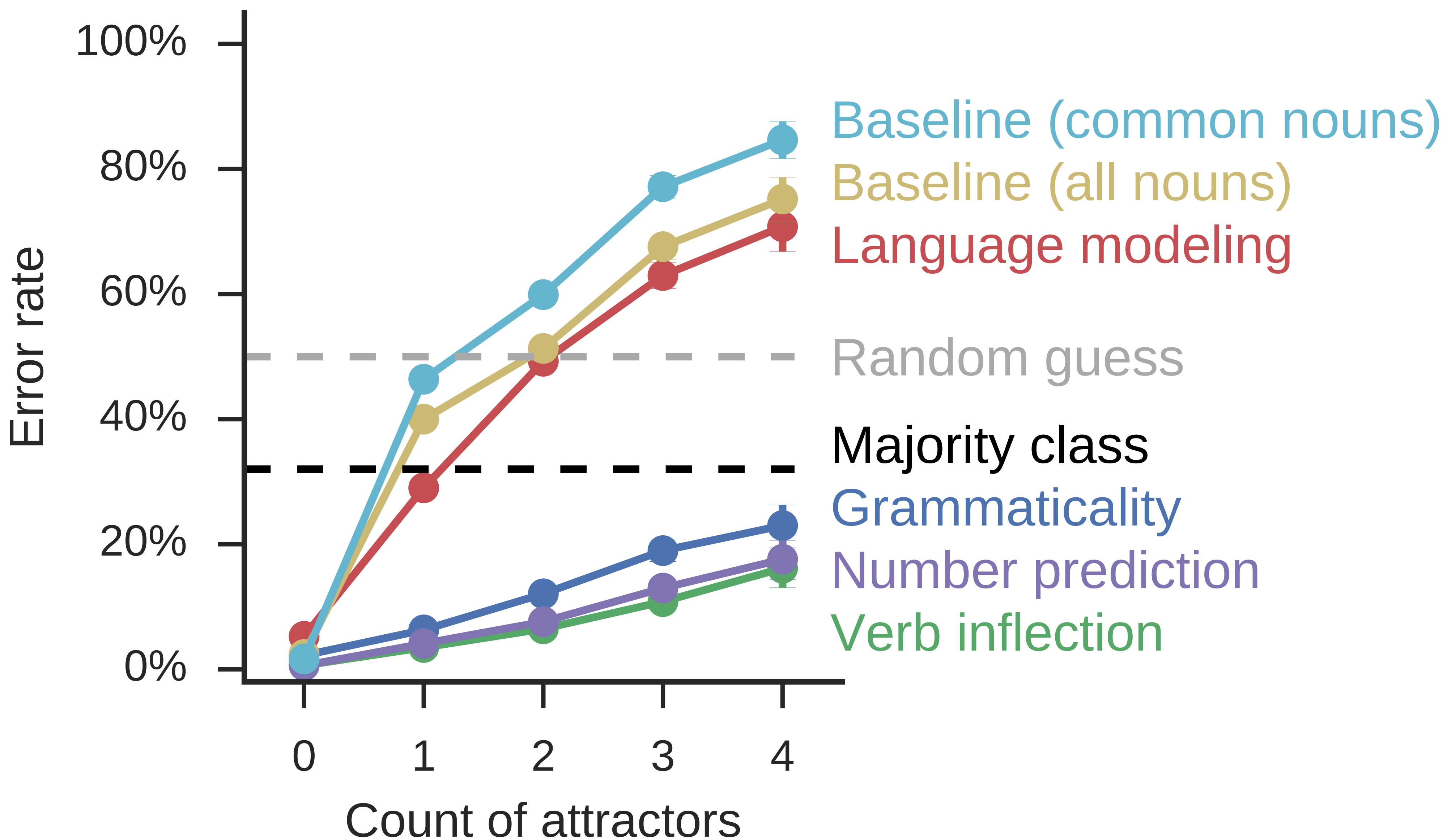
LSTMs learn agreement remarkably well.

but we trained it on the agreement task.

does a language model learn agreement?

Can a sequence LSTM learn agreement?

does a language model learn agreement?



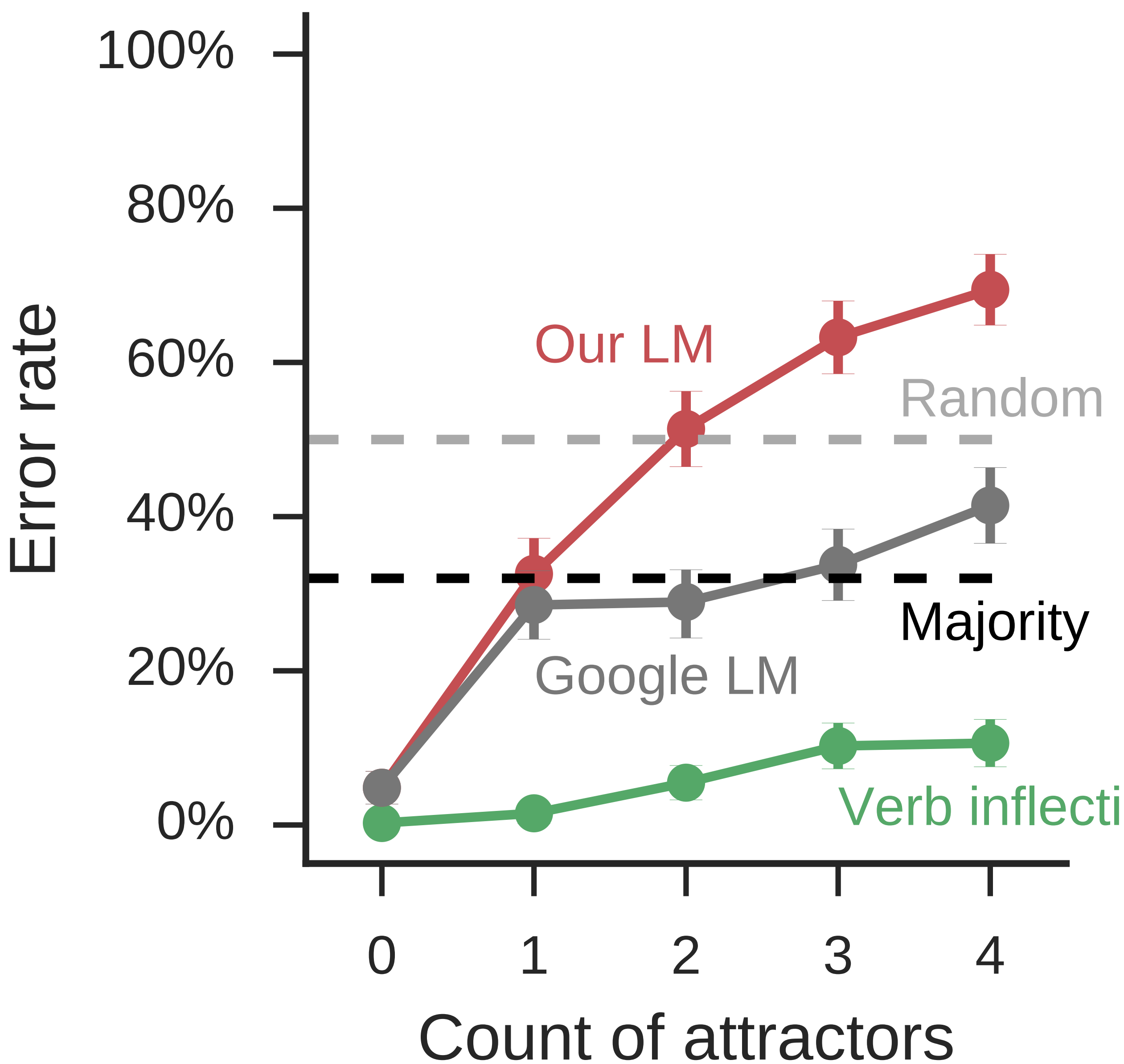
Can a sequence LSTM learn agreement?

does a language model learn agreement?

what if we used the **best LM in the world?**

Can a sequence LSTM learn agreement?

does a language model learn agreement?



Google's beast LM does better than ours but still struggles considerably.

Can a sequence LSTM learn agreement?

does a language model learn agreement?

LSTMs can learn agreement very well.

But LSTM-LM **does not** learn agreement.

Explicit error signal is required.

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 <u>article</u> 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.

Agreement Prediction -- What's next

- Many ways to extend this:
  - More languages
  - More phenomena
  - Make it fail!
      - and then improve it.

what do trained LSTM acceptors encode?

Extracting FSAs from RNNs

B I U
N L P

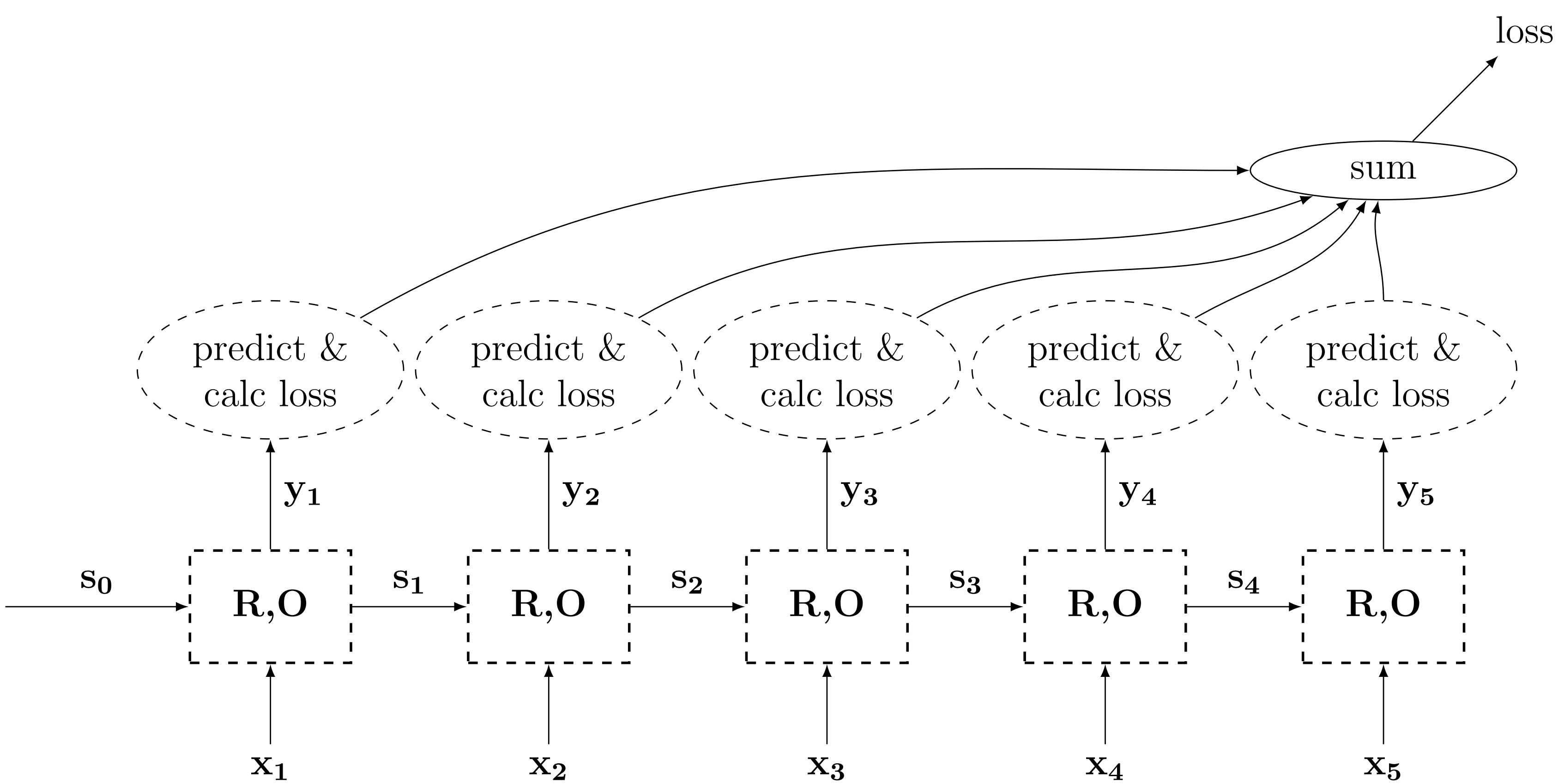
TECHNION
Israel Institute
of Technology

Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples

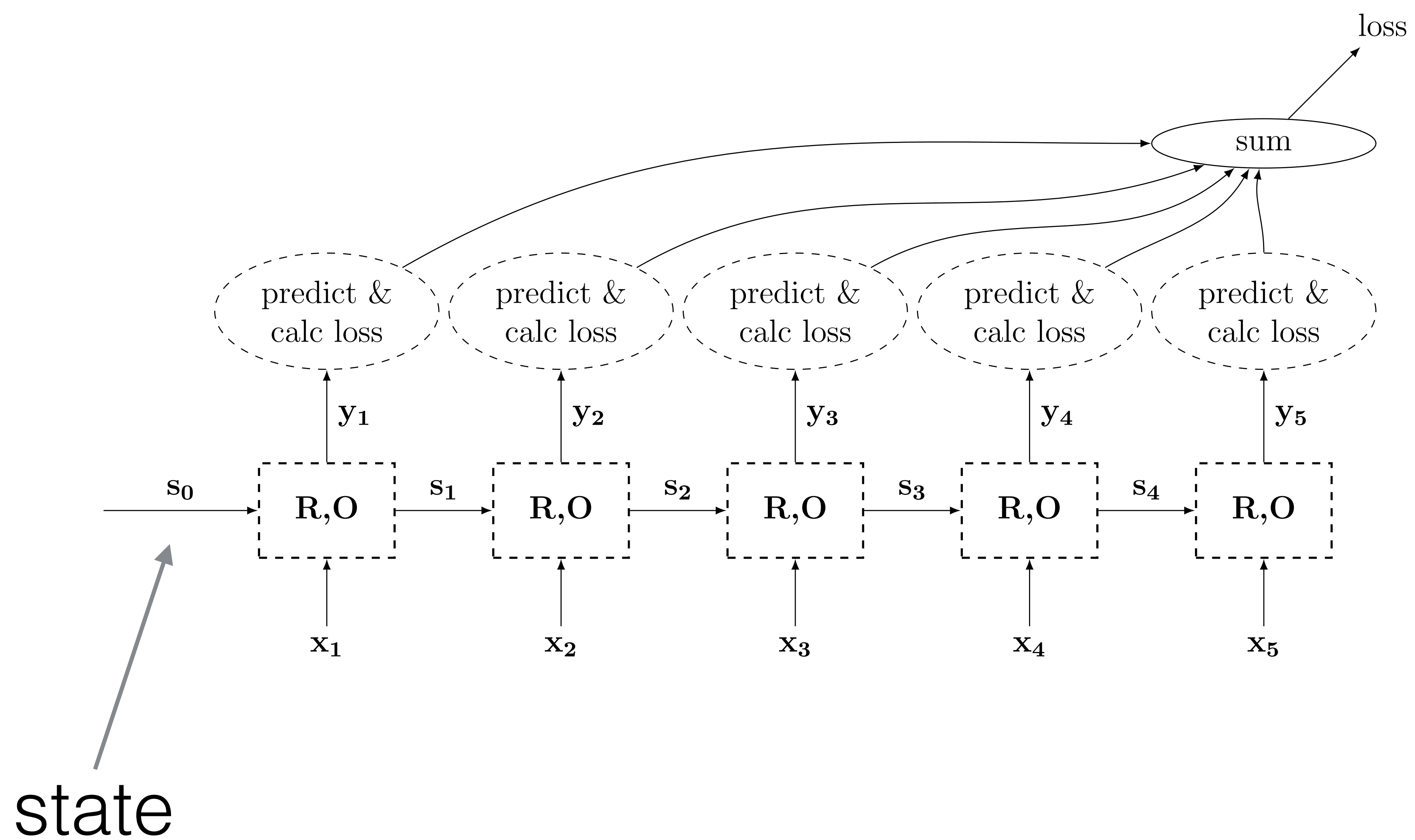
Gail Weiss[1], Yoav Goldberg[2], and Eran Yahav[1]

RNN acceptors as State Machines

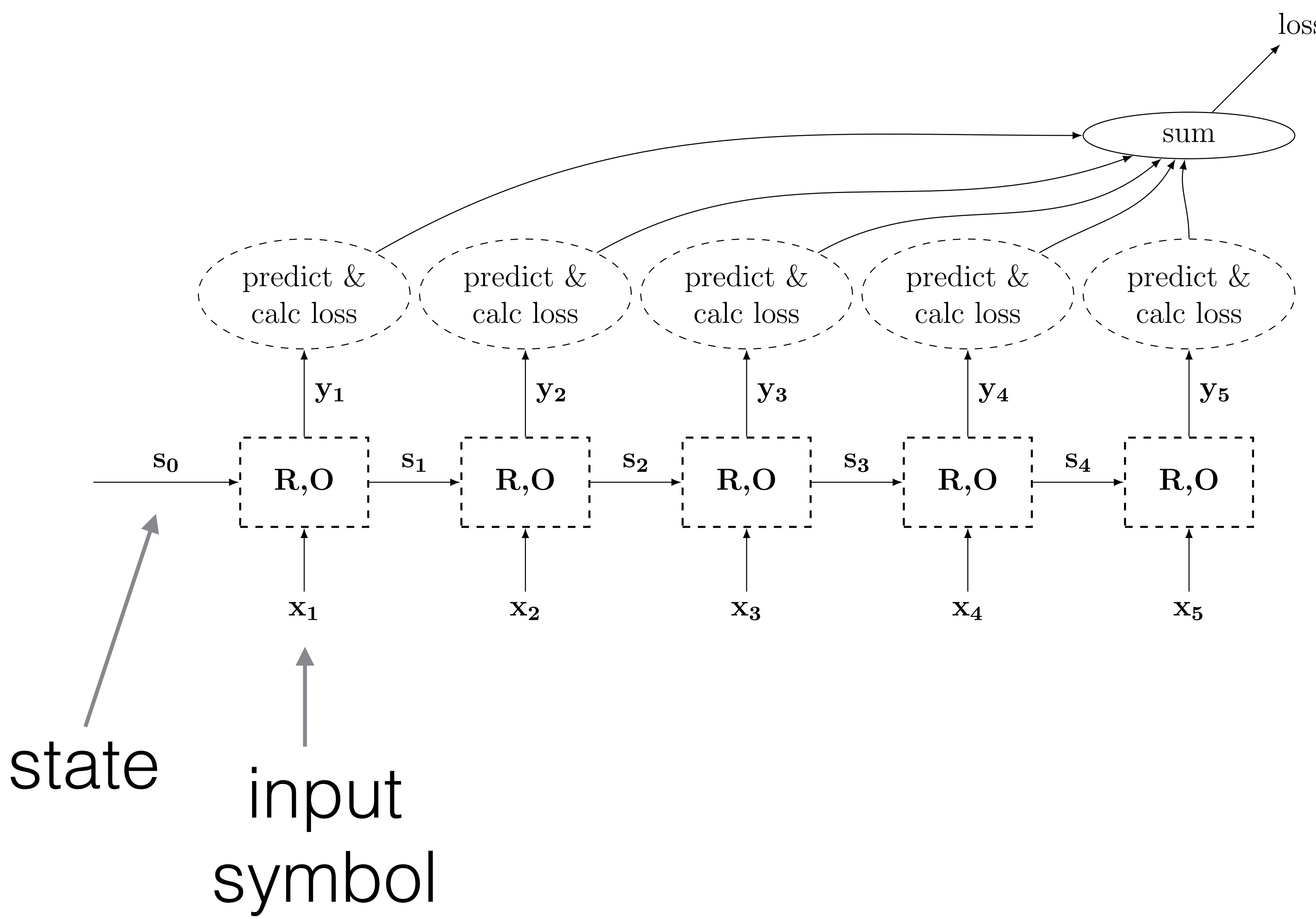
RNN acceptors as State Machines



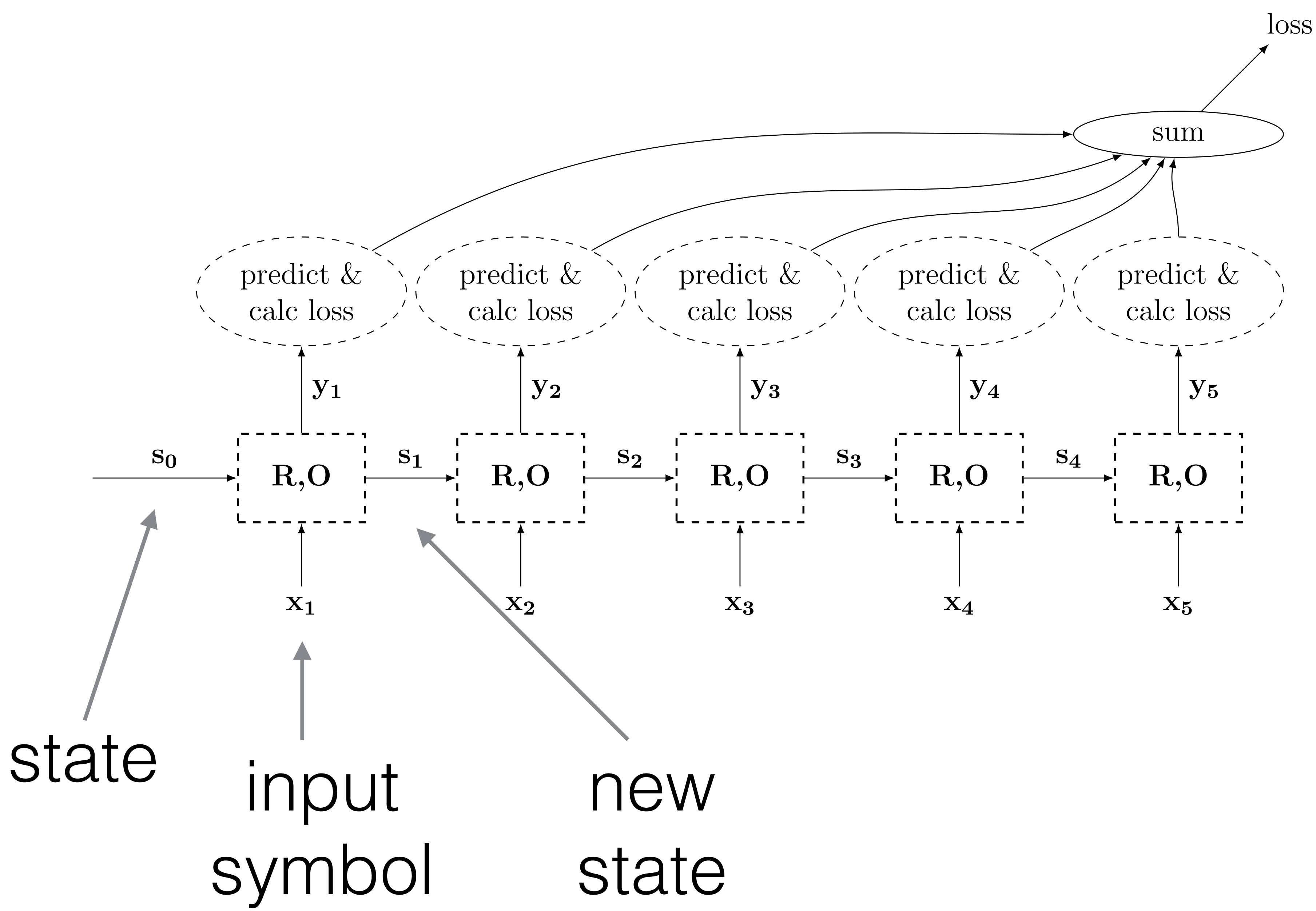
RNN acceptors as State Machines



RNN acceptors as State Machines

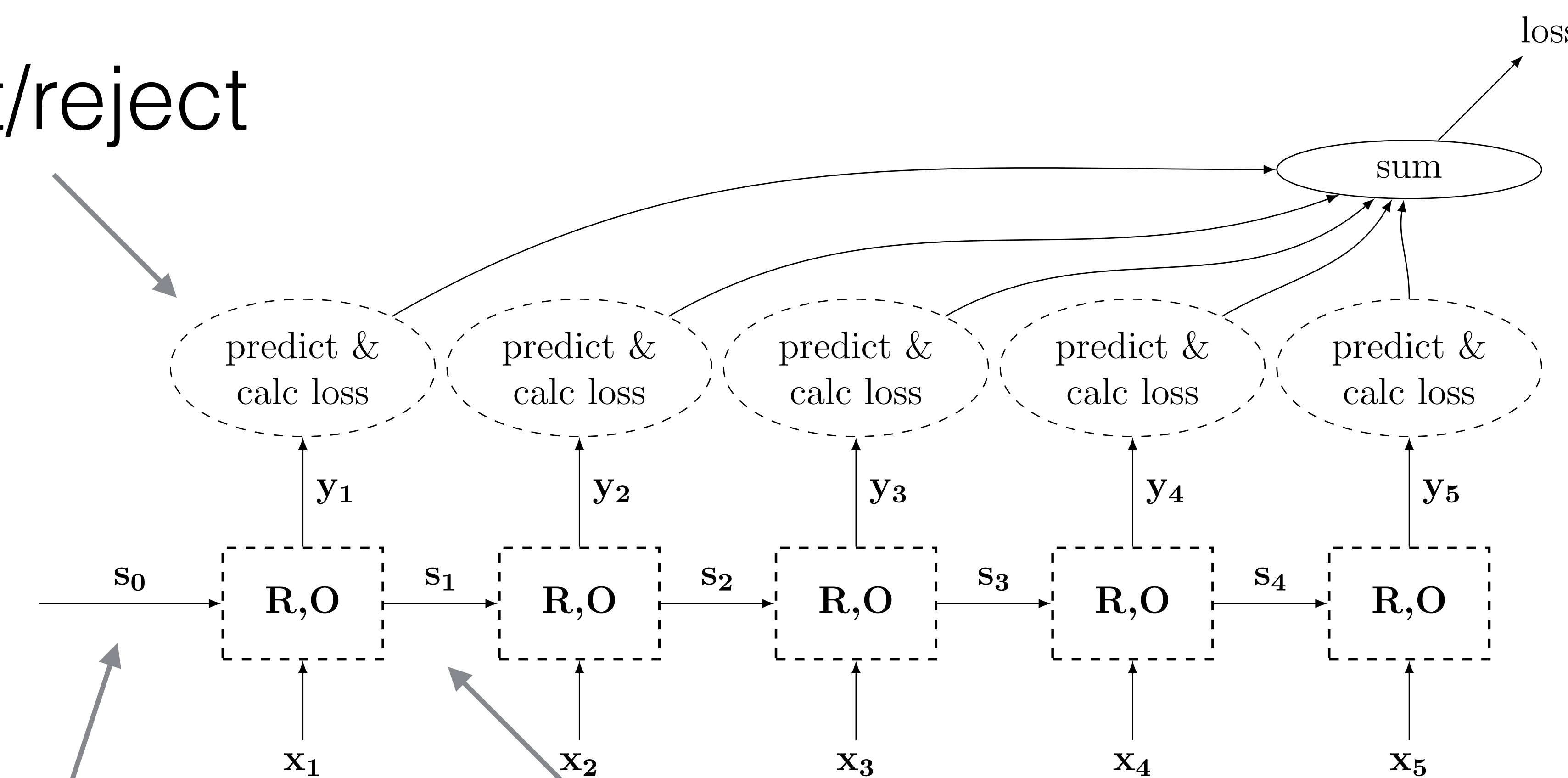


RNN acceptors as State Machines



RNN acceptors as State Machines

accept/reject



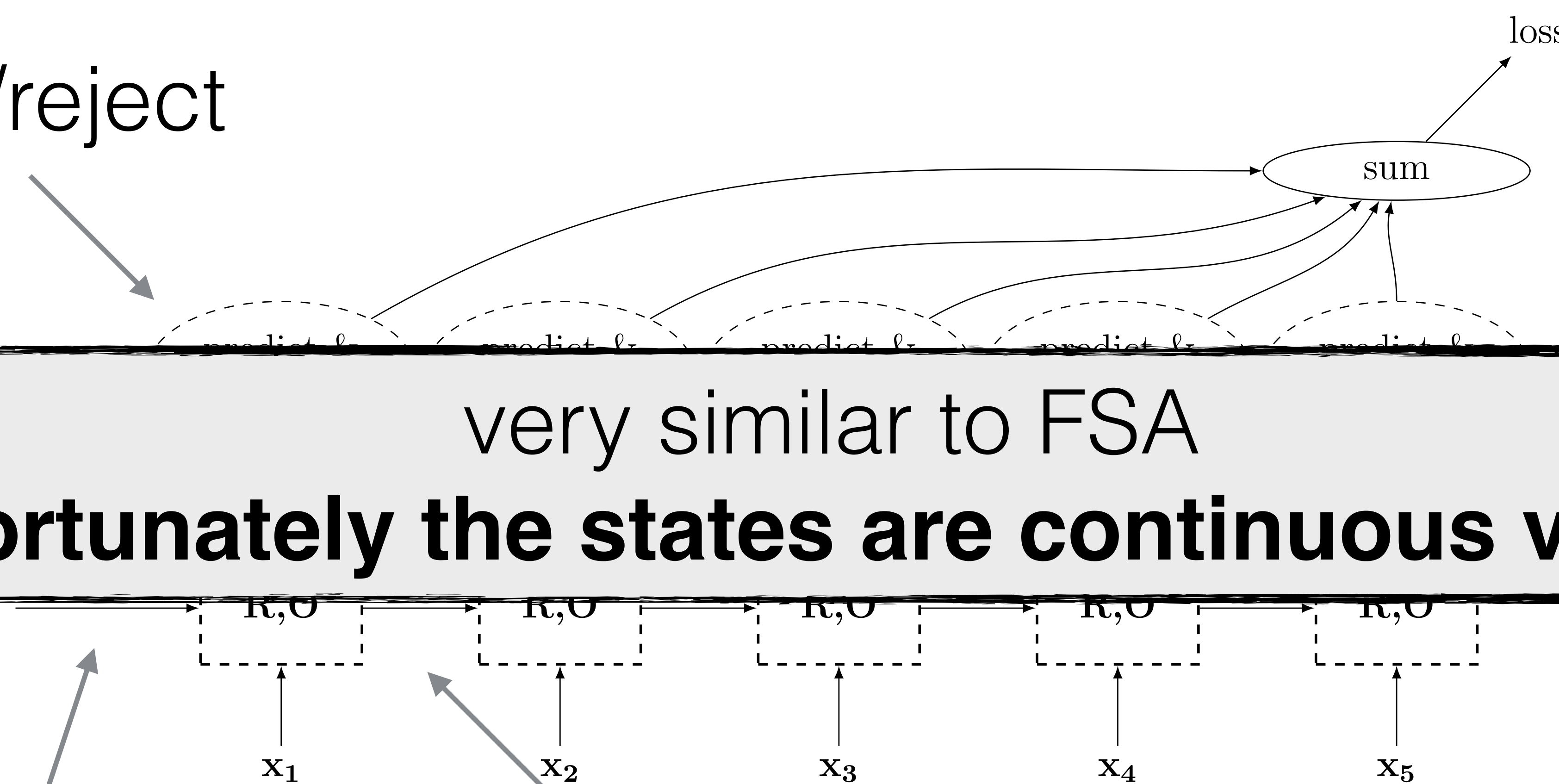
state

input symbol

new state

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

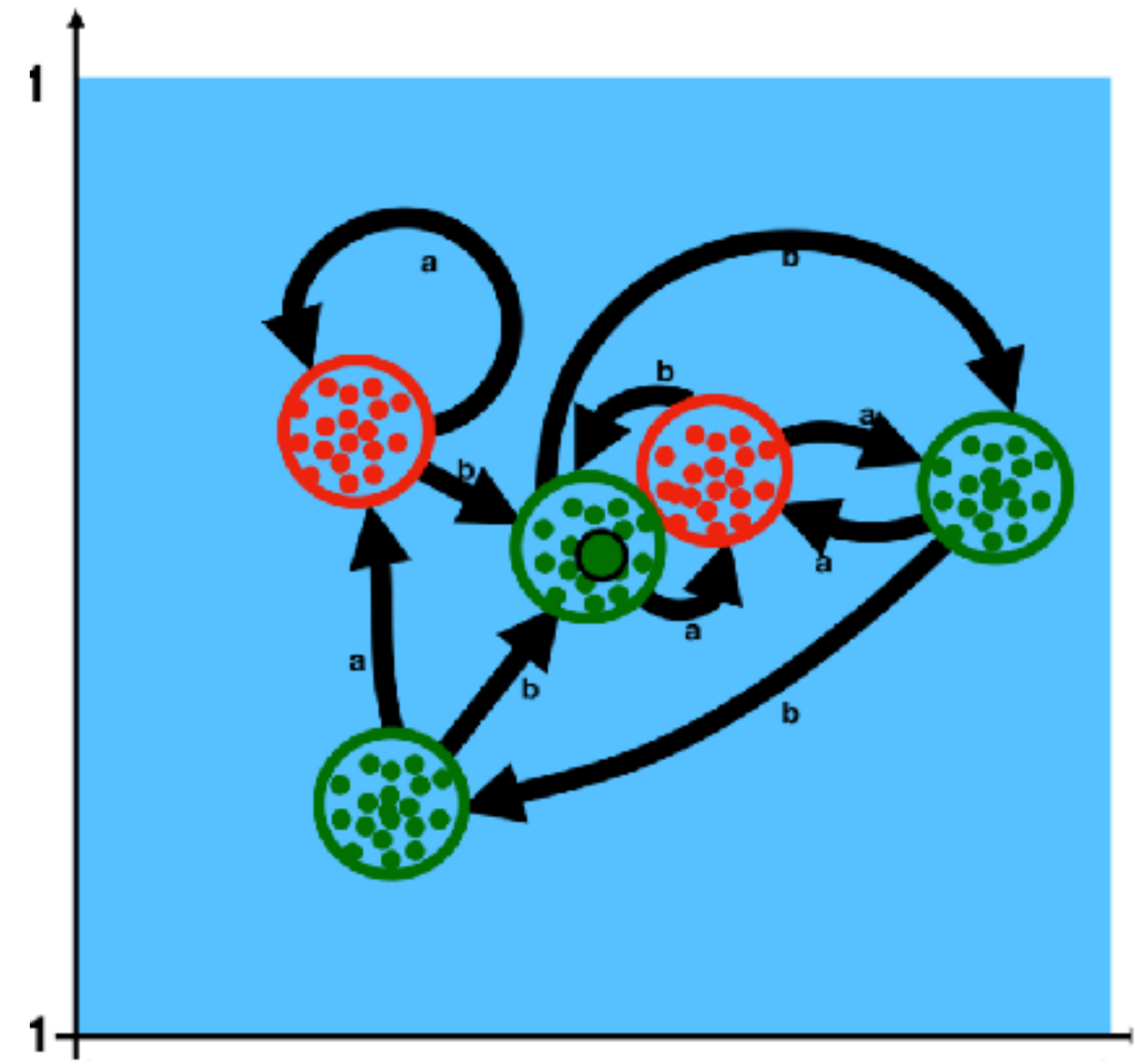
Easy. Just run the word through the RNN.

Equivalence Queries

Hard. Requires some trickery.

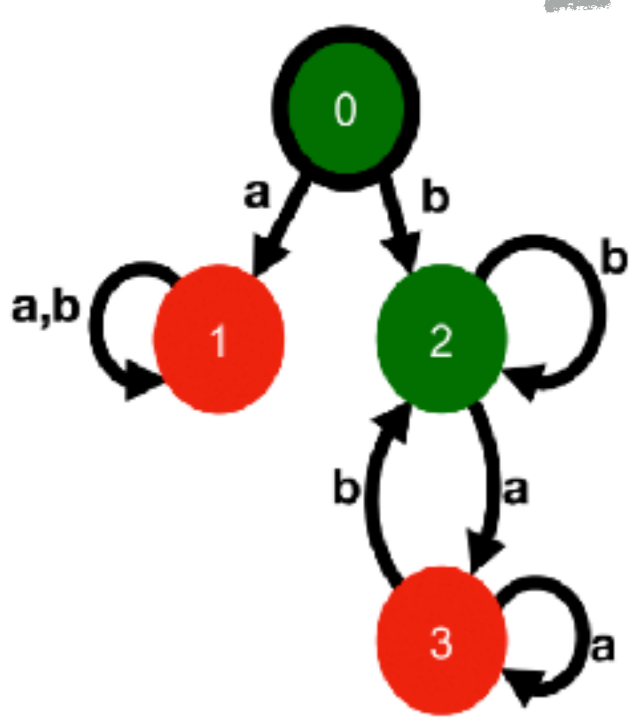
Answering Equivalence Queries

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

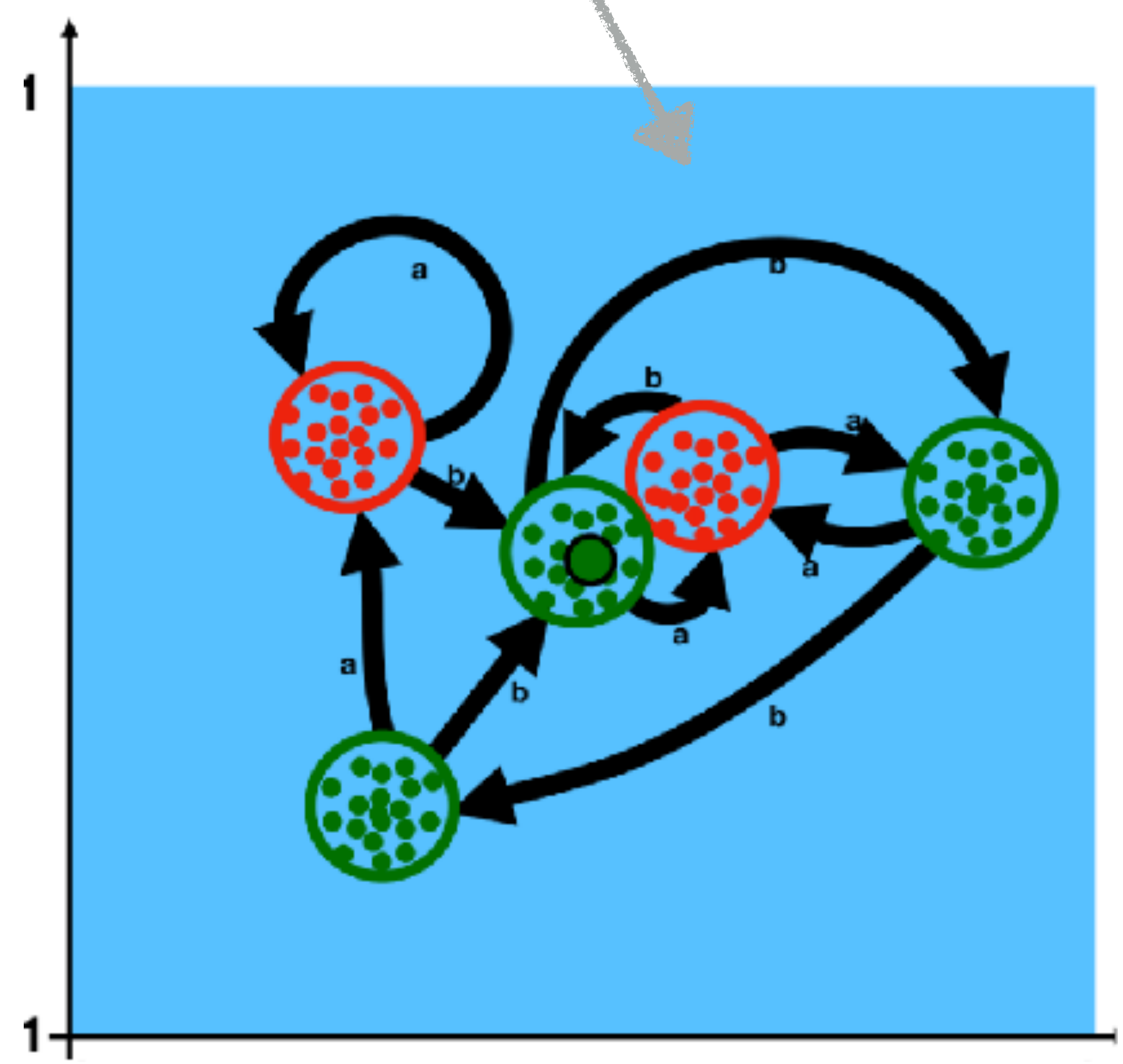


Answering Equivalence Queries

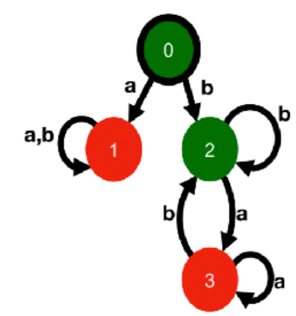
- Compare L* **Query FSA** to **RNN-Abstract-FSA**.

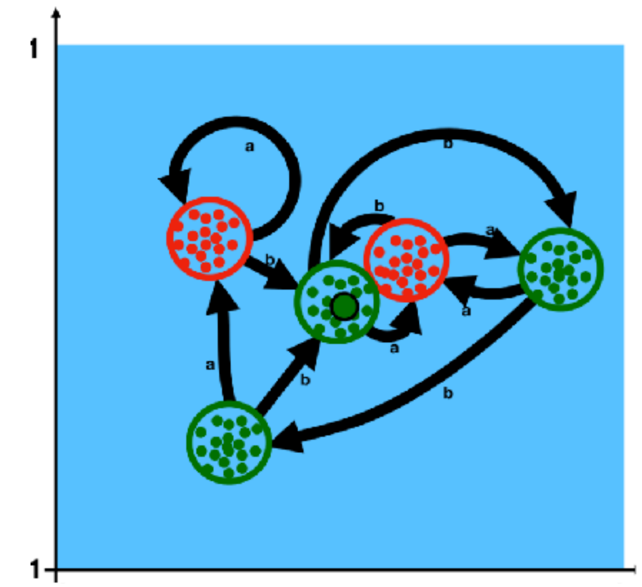


=??



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

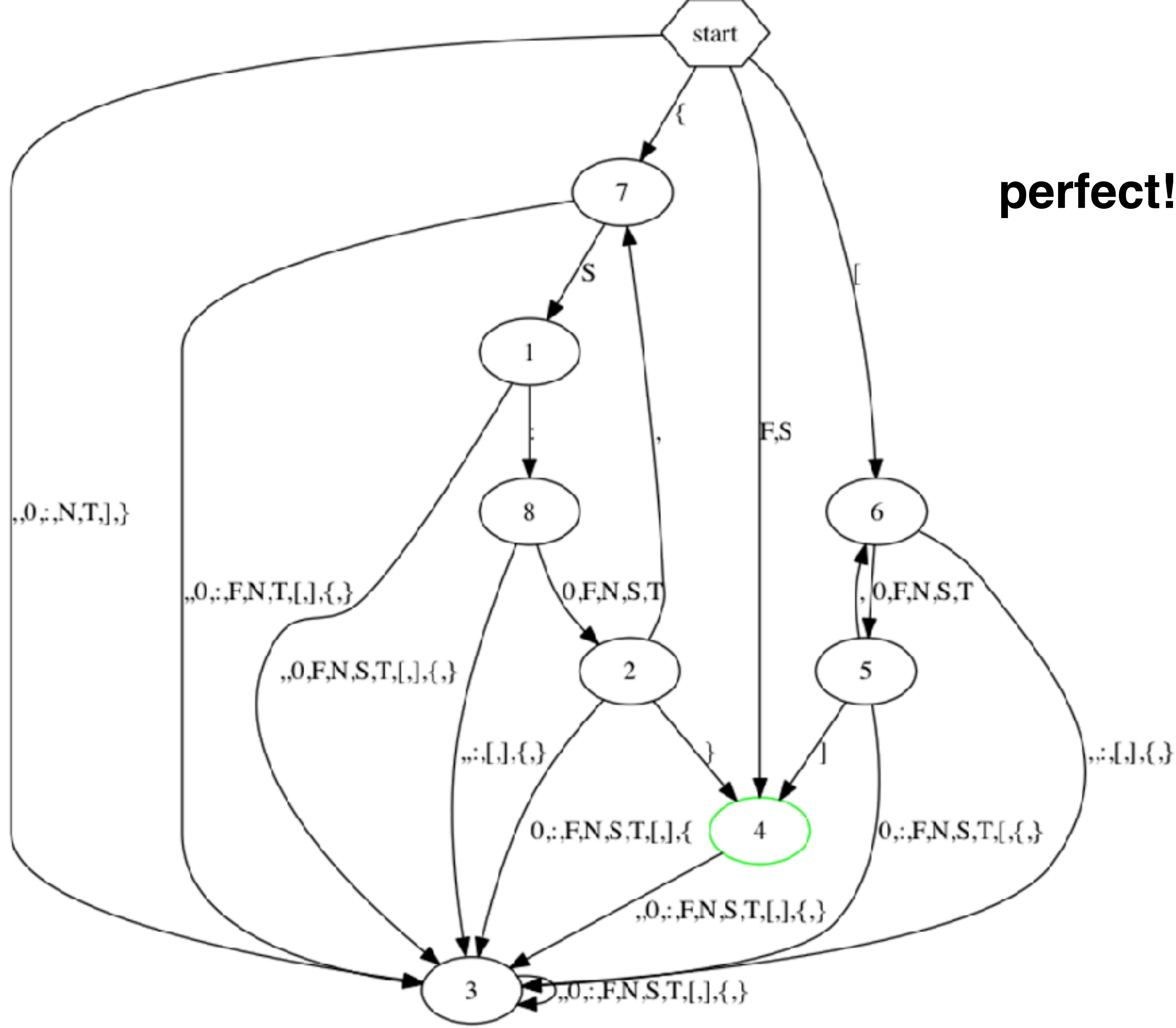
- **Many random FSAs:**
    - 5 or 10 states, alphabet sizes of 3 or 5
- LSTM/GRU with 50, 100, 500 dimensions.
- The FSAs were **learned well** by LSTM / GRU
- And **recovered well** by L*.

"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 , : { } [ ]`

perfect!



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,[,],{,}

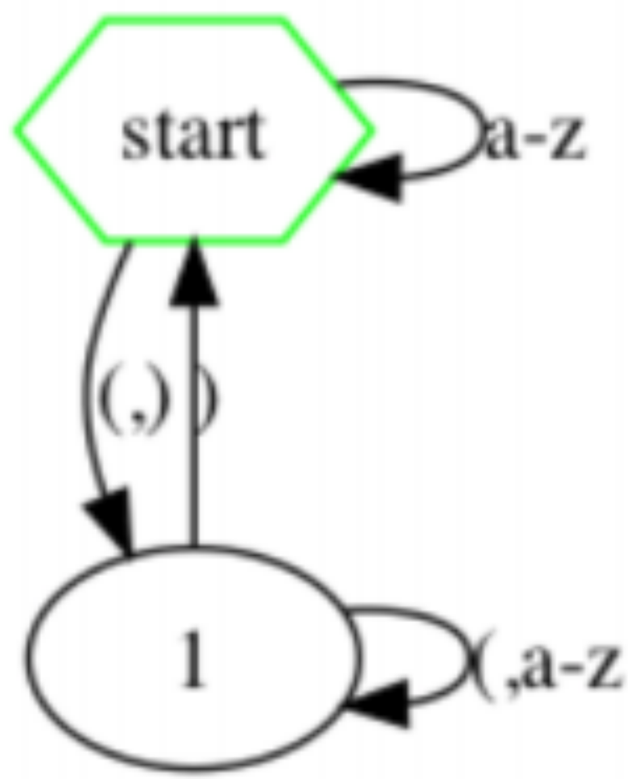
Balanced Parenthesis

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

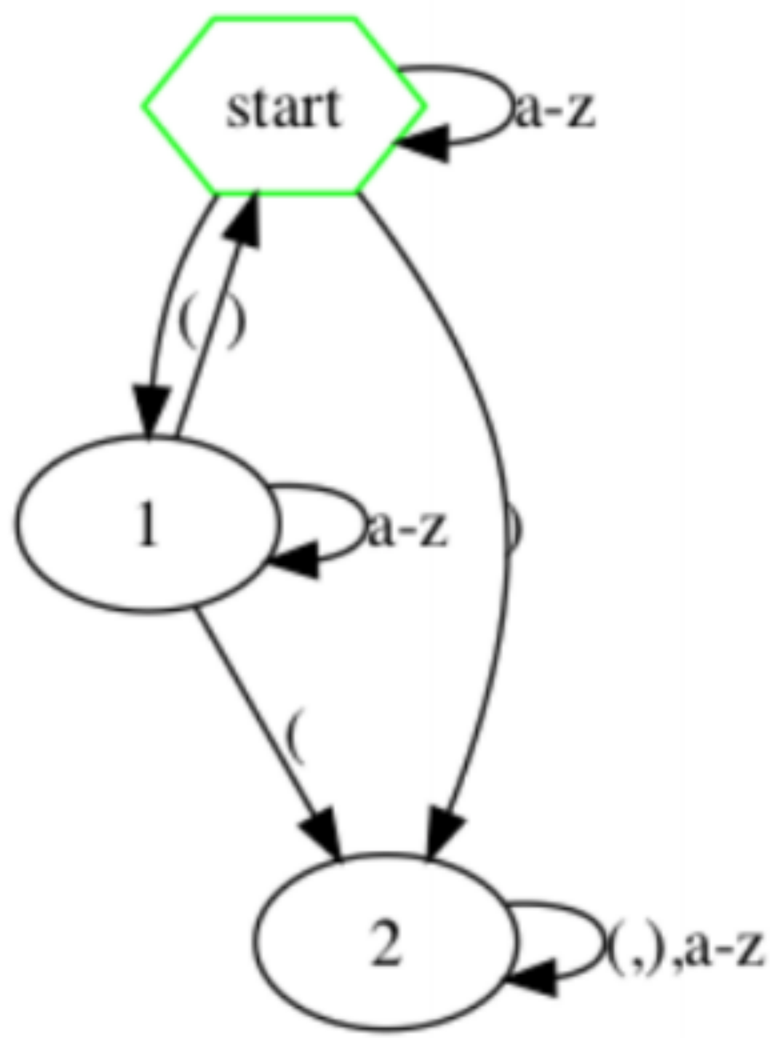
alphabet: `a-z ( )`

nesting level up to 8.

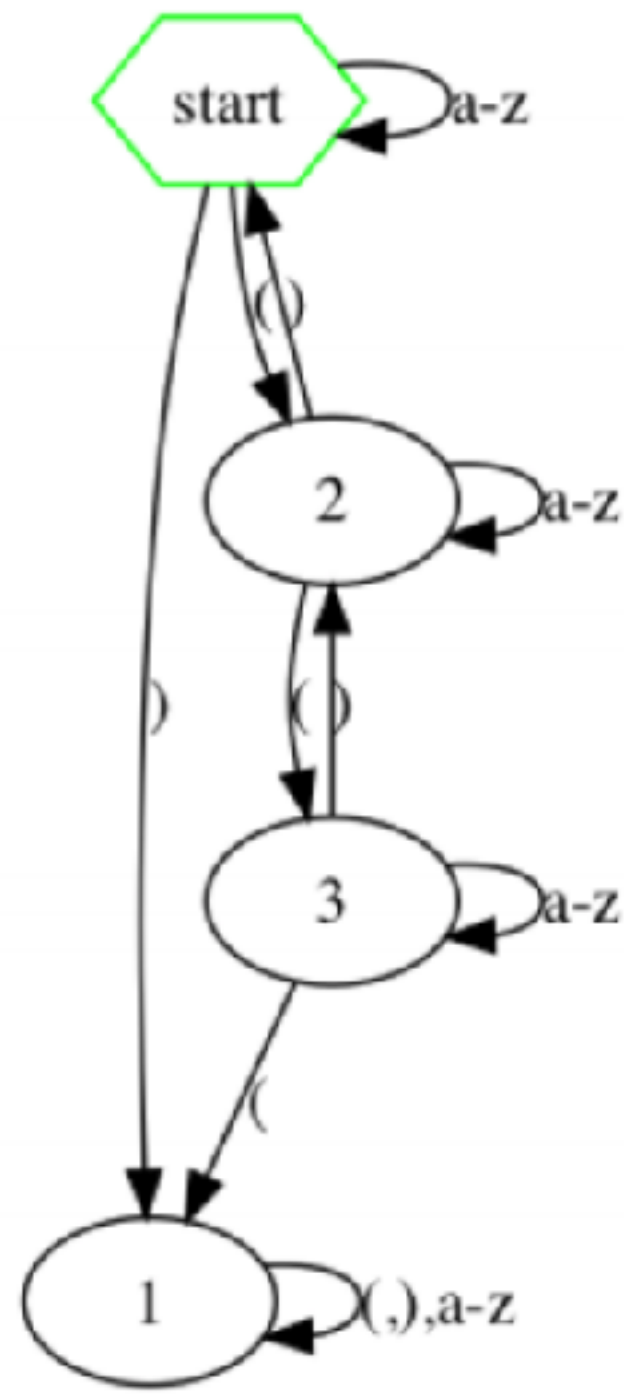
Balanced Parenthesis



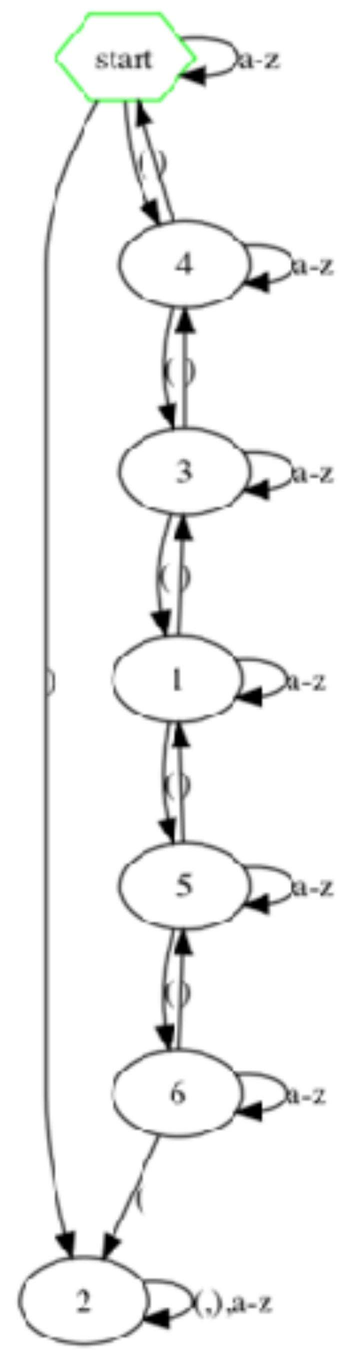
Balanced Parenthesis



Balanced Parenthesis



Balanced Parenthesis

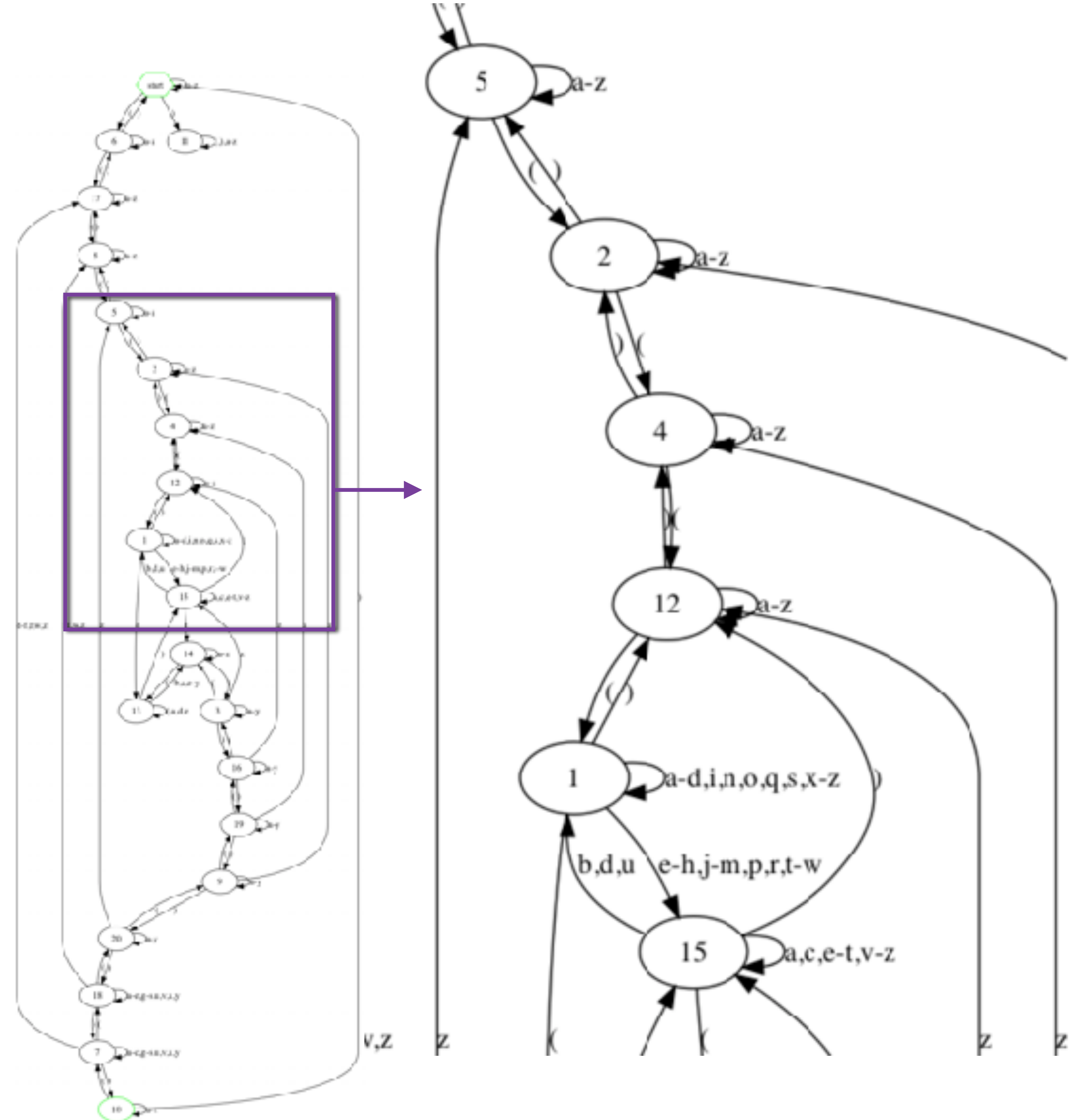


Balanced Parenthesis

final automaton:

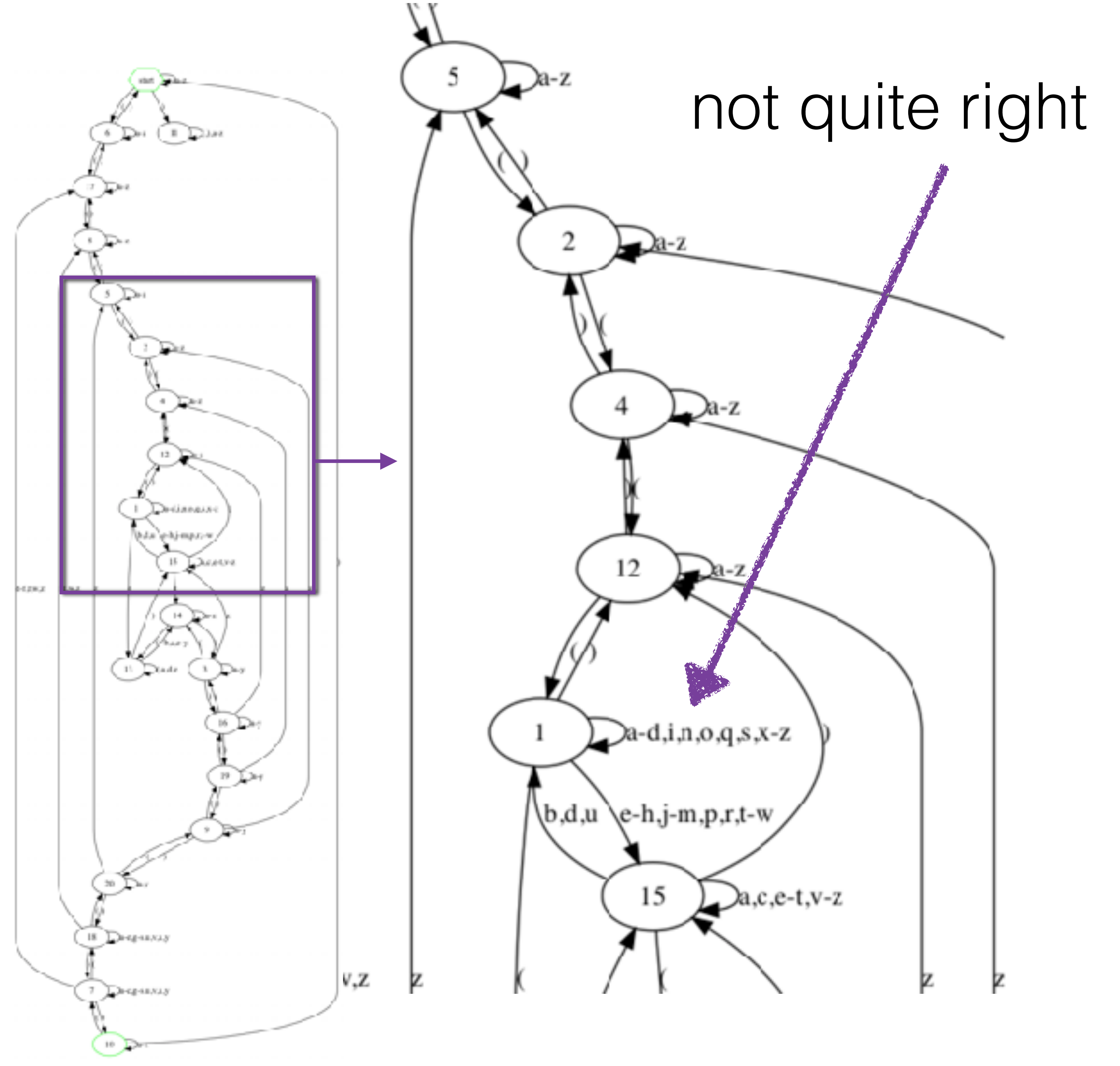
Balanced Parenthesis

final automaton:



Balanced Parenthesis

final automaton:



"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.

some 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 (the talk)

- LSTM are very powerful
    - We know how to use them.
    - We don't know enough about their power and limitations.
    - We should try to understand them better.

Understanding LSTMs

- **Our humble start**
  - Experiments for understanding sentence representations.
  - LSTMs and English subject-verb agreement.
  - Extracting FSAs from trained LSTMs.
- **Still much to do. Help us do it.**

thanks for listening