Trying to Understand Recurrent Neural Networks for Language Processing

Yoav Goldberg

Blackbox NLP workshop, 2018

BIU NLP

Bar-Ilan University

ALLEN INSTITUTE for ARTIFICIAL INTELLIGENCE

My Research

Core Building Blocks for NLP

My Research

Core Building Blocks
for NLP

Using Machine Learning

Trying to Understand Recurrent Neural Networks for Language Processing

Trying to Understand Recurrent Neural Networks for Language Processing

NLP

Trying to Understand Recurrent Neural Networks for Language Processing

ML

NLP

GAP!

Trying to Understand Recurrent Neural Networks for Language Processing

ML

NLP

How do we do NLP?

- 1950 -- ~1990s ---> Write many rules
- 1990s -- ~2000s ---> Corpus based statistics
- 2000s -- ~2014 ---> Supervised machine learning
- 2014 -- today ---> "deep learning"

How do we do NLP?

- 1950 -- ~1990s ---> Write many rules **<-- transparent**
- 1990s -- ~2000s ---> Corpus based statistics
- 2000s -- ~2014 ---> Supervised machine learning
- 2014 -- today ---> "deep learning" **<-- BlackBoxNLP**

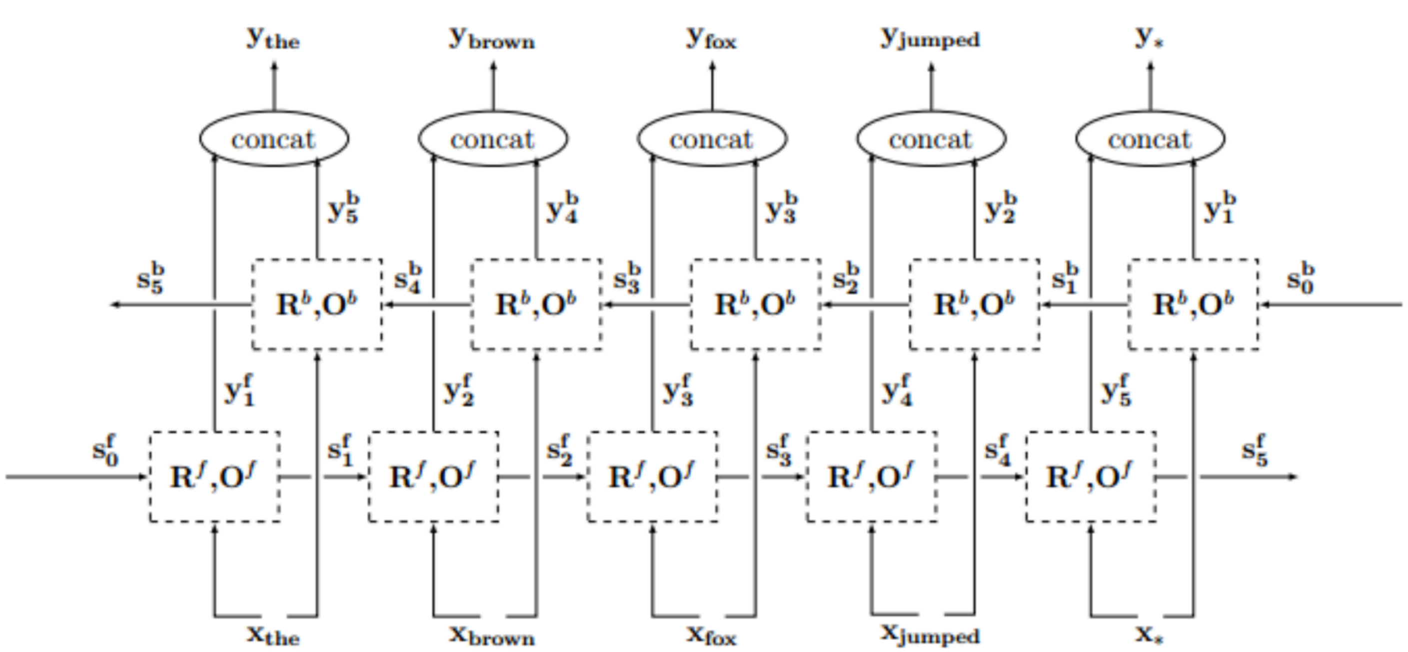
How do we do NLP?

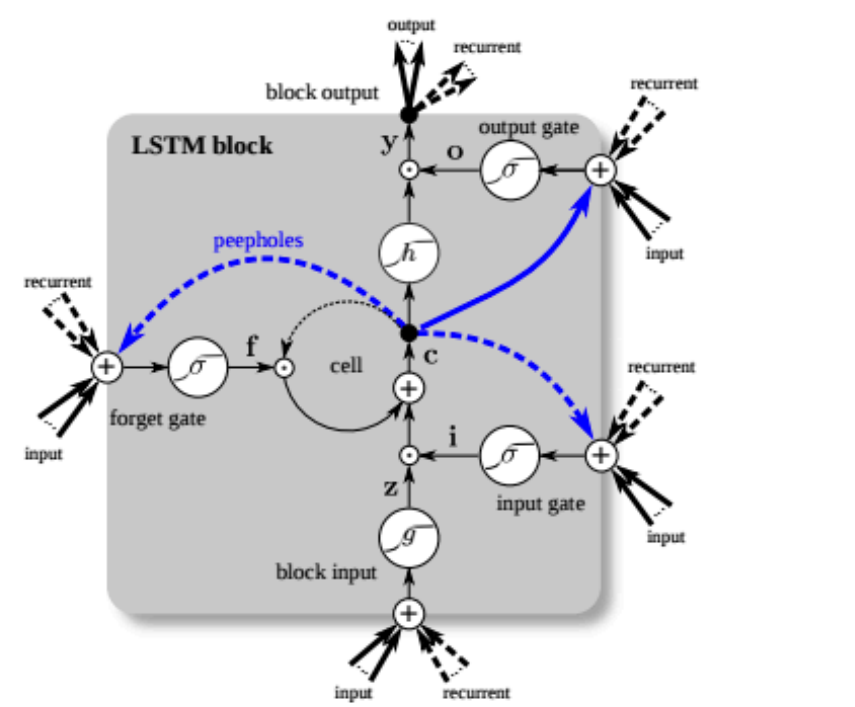
- 1950 -- ~1990s ---> Write many rules **<-- transparent**
- 1990s -- ~2000s ---> Corpus based statistics
- 2000s -- ~2014 ---> Supervised machine learning
- 2014 -- today ---> "deep learning" **<-- BlackBoxNLP**
- 2021+ ---> write rules, aided by ML/DL

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

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

Doing stuff with LSTMs

LSTMs are very capable learners

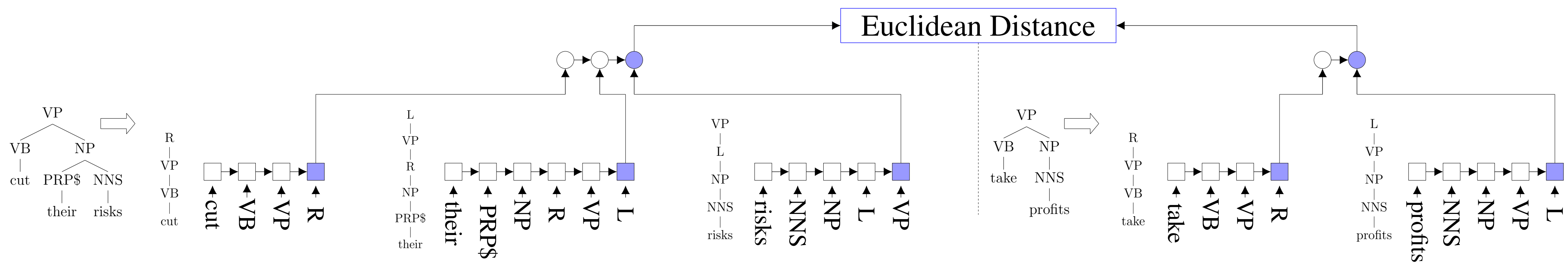
Use them to build stuff

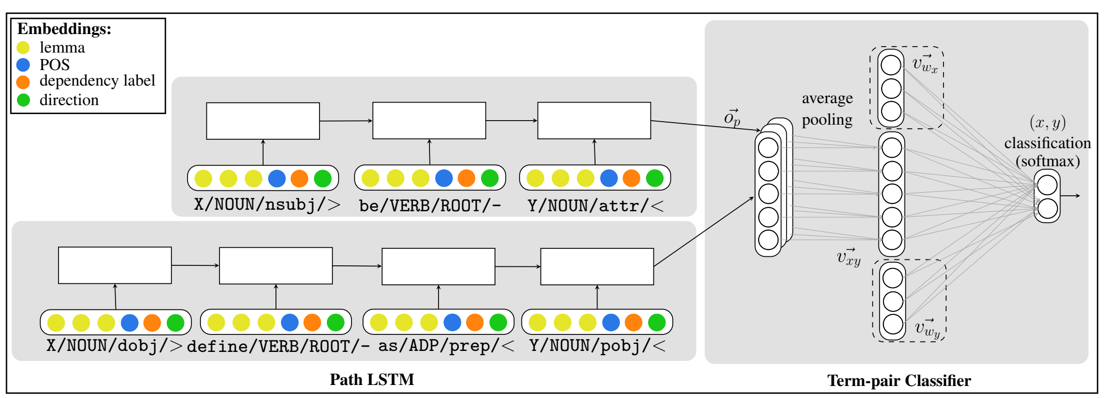
strong results

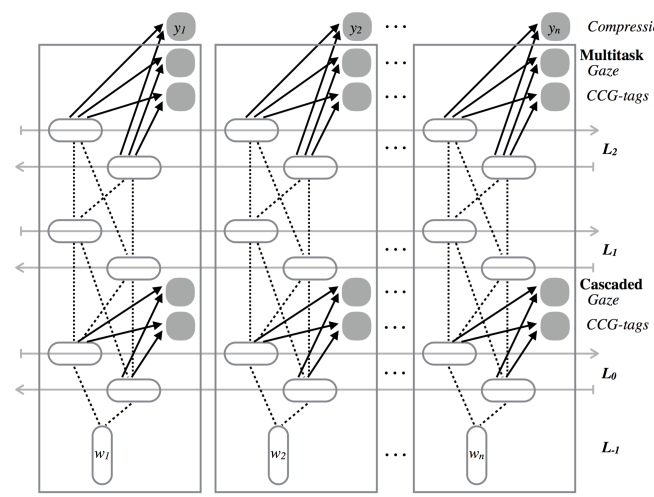
make reviewers happy

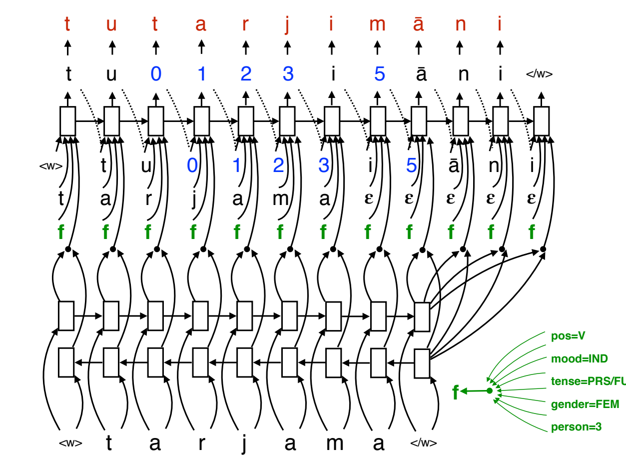
publish many papers

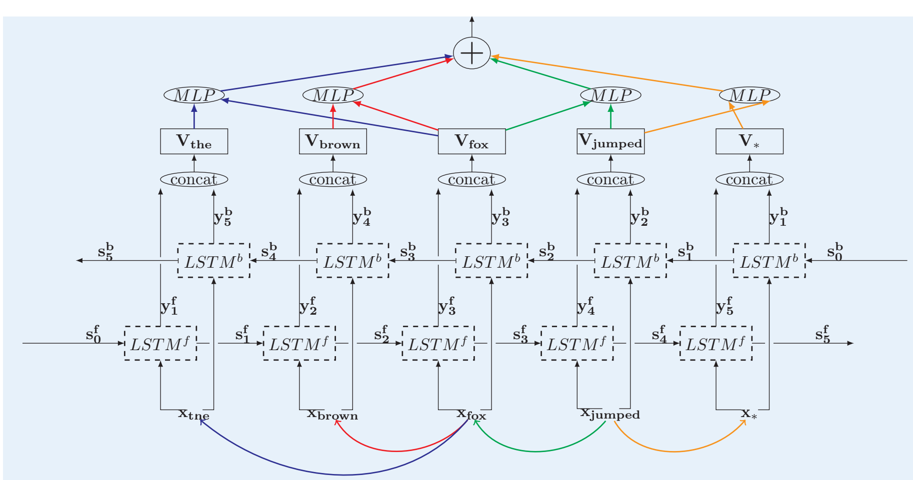
Doing stuff with LSTMs

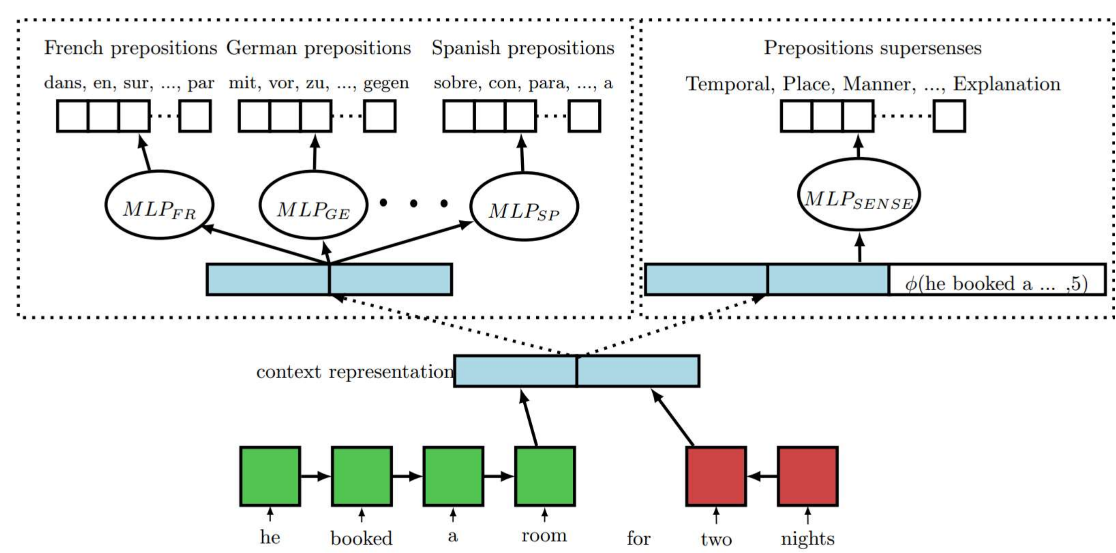




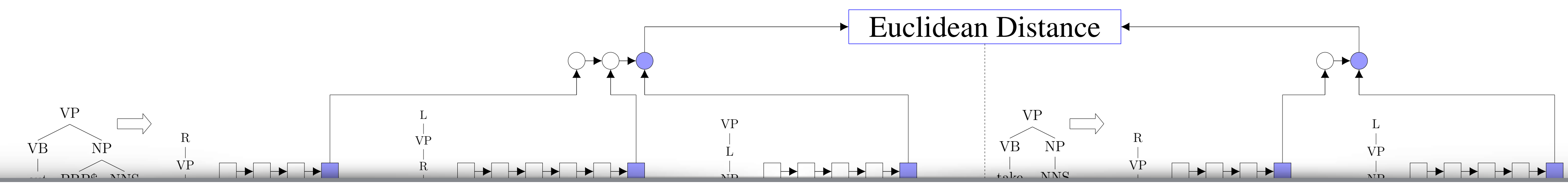


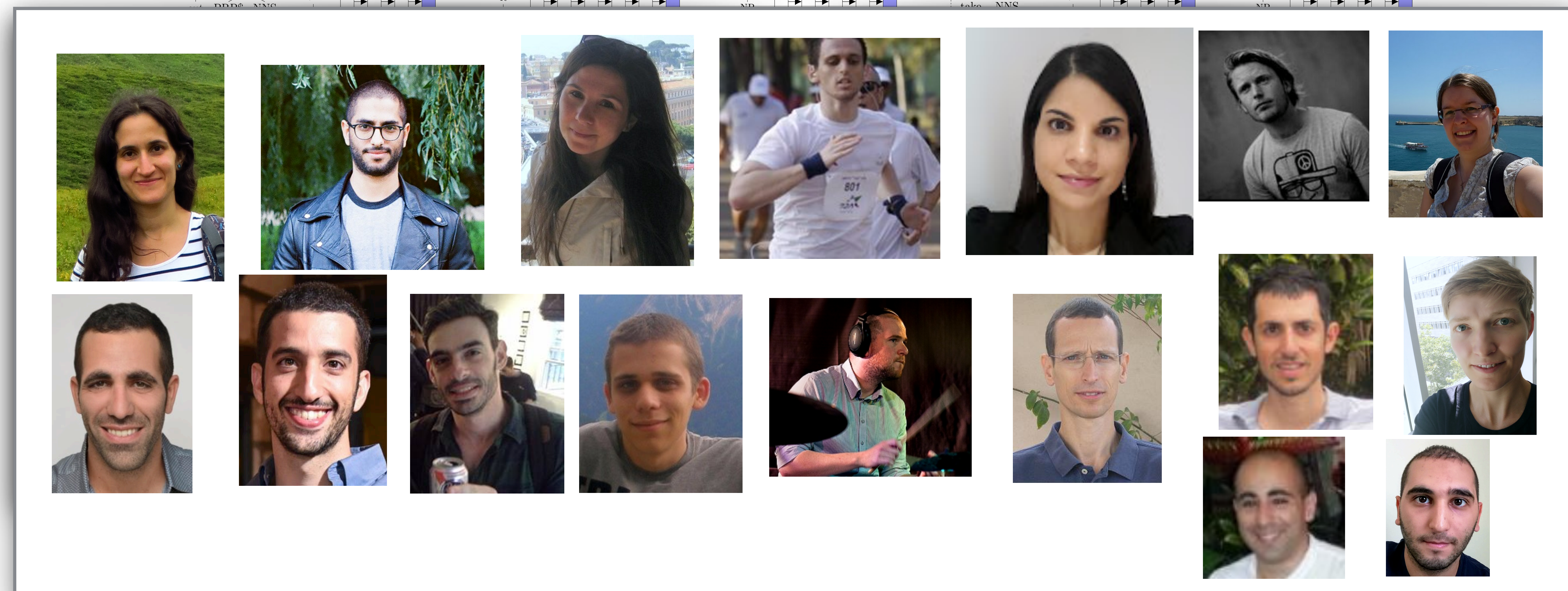


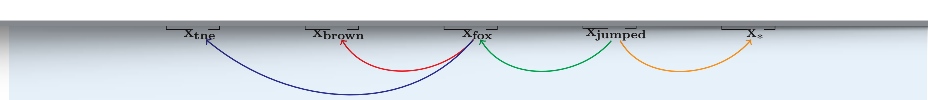


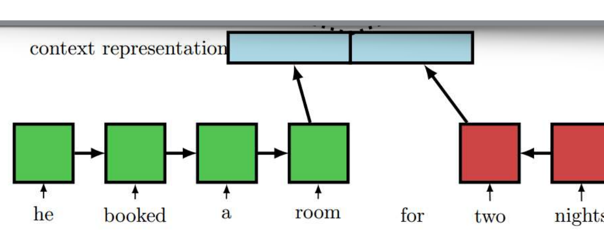


Doing stuff with LSTMs









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

B I U
N L P

On-the-fly Operation Batching in Dynamic Computation Graphs

Graham Neubig*
Language Technologies Institute
Carnegie Mellon University
gneubig@cs.cmu.edu

Yoav Goldberg*
Computer Science Department
Bar-Ilan University
yogo@cs.biu.ac.il

Chris Dyer
DeepMind
cdyer@google.com

build tools to build stuff

build stuff faster

help others build stuff

publish more papers

∂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

Doing stuff with LSTMs

LSTMs are very capable learners

Use them to build stuff

Try to understand them

Except for this awesome workshop!
things are changing?

scratching the surface

reviewers don't care much

I find it really interesting

Doing stuff with LSTMs

Alexander Clark and 2 others liked

Jelle Zuidema @wzuidema · Sep 6

Replying to @mdlhx

Yes, the **#BlackboxNLP** program looks great. And kudos to its organizers for allowing #EMNLP to colocate with it!

1 3 14

Except for this awesome workshop!
things are changing?

scratching the surface

reviewers don't care much

I find it really interesting

Understanding LSTMs

Q1: What is encoded/captured in a vector?

Understanding LSTMs

Q1: 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]



Understanding LSTMs

Q1: 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?

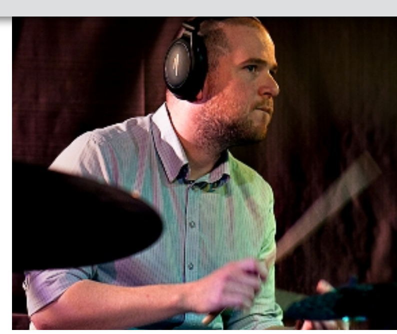
Understanding LSTMs

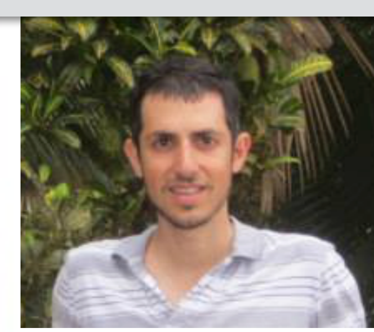
Q1: What is encoded/captured in a vector?

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?





Understanding LSTMs

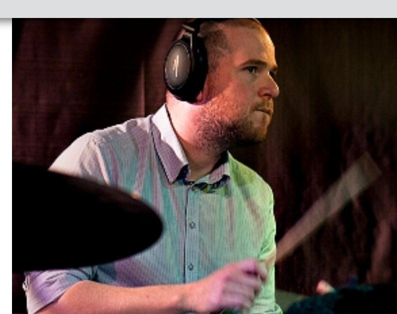
Q1: What is encoded/captured in a vector?

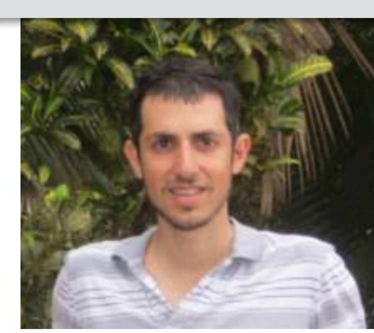
Published as a conference paper at ICLR 2017

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

Understanding LSTMs

Q1: What is encoded/captured in a vector?

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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: 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

Understanding LSTMs

Q1: What is encoded/captured in a vector?

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

Understanding LSTMs

Q1: 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**

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

Understanding LSTMs

Q1: 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 works in this workshop!

Understanding LSTMs

Q1: 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 works in this workshop!

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

is top-tier NLP too conservative?

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

Understanding LSTMs

Q2: 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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

many works in this workshop!

Understanding LSTMs

Q2: what kinds of linguistic structures can be captured by an RNN?

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

many works in this workshop!

Including our poster on Basque

Understanding LSTMs

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

Representation of Linguistic Form and Function in Recurrent Neural Networks

Ákos Kádár*
Tilburg University

Grzegorz Chrupała*
Tilburg University

Afra Alishahi*
Tilburg University

pioneering work in this space

(also took forever to get accepted)

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

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

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

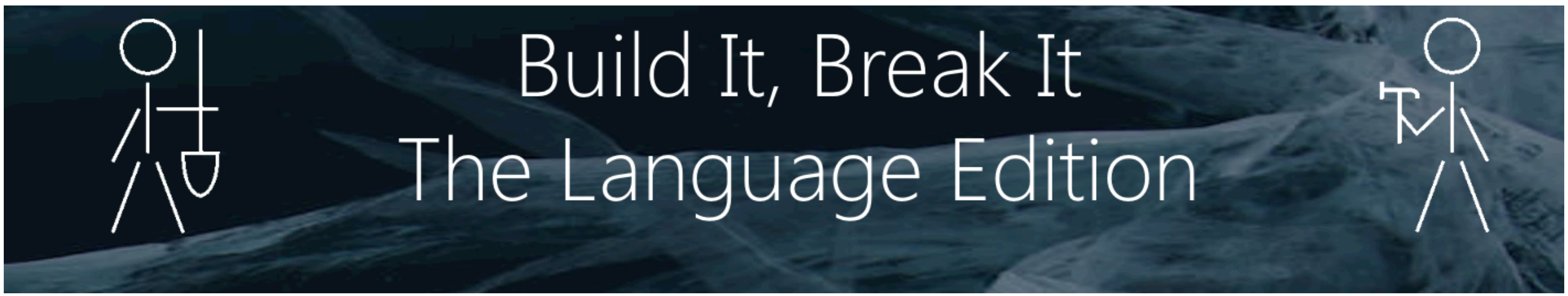
my student Alon Jacovi will present our work on analyzing **1D-CNNs** for text.

Understanding LSTMs

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

Understanding LSTMs

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



Understanding LSTMs

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

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

Understanding LSTMs

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

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

and others from other groups

Q1: What is encoded/captured in a vector?

Q2: what kinds of linguistic structures
can be captured by an RNN?

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

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

Q1: What is encoded/captured in a vector?

Q2: what kinds of linguistic structures
can be captured by an RNN?

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

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

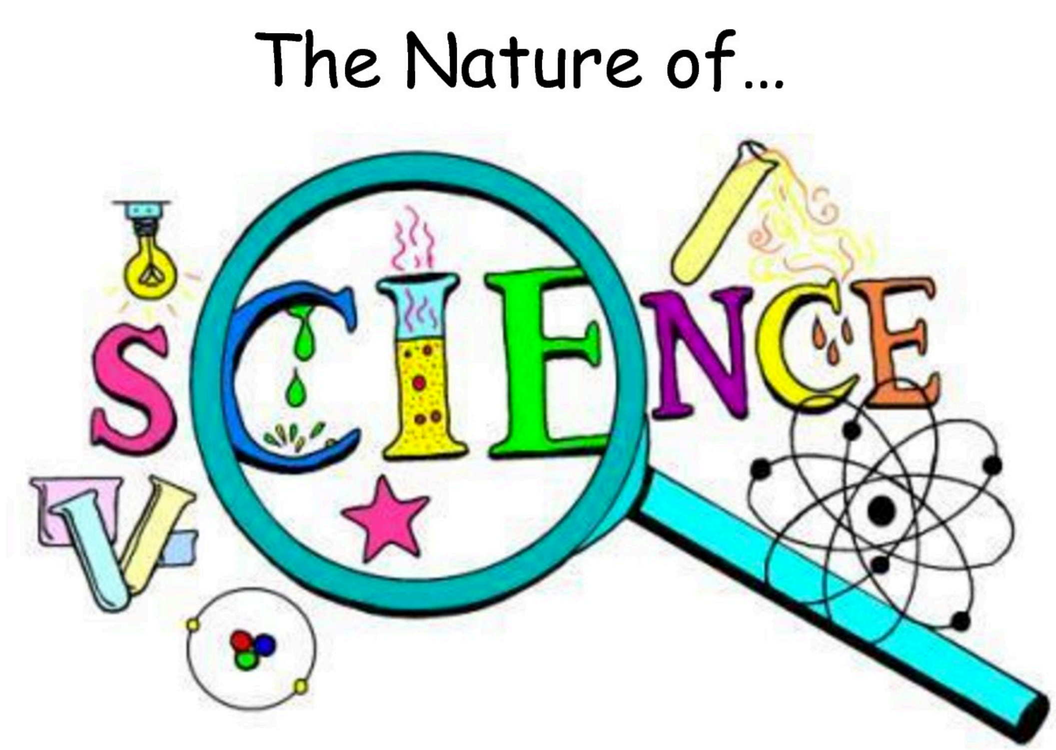
The Nature of...

Q1: What is encoded/captured in a vector?

Q2: what kinds of linguistic structures
can be captured by an RNN?

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

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



Treat the representations / model as an "organism".

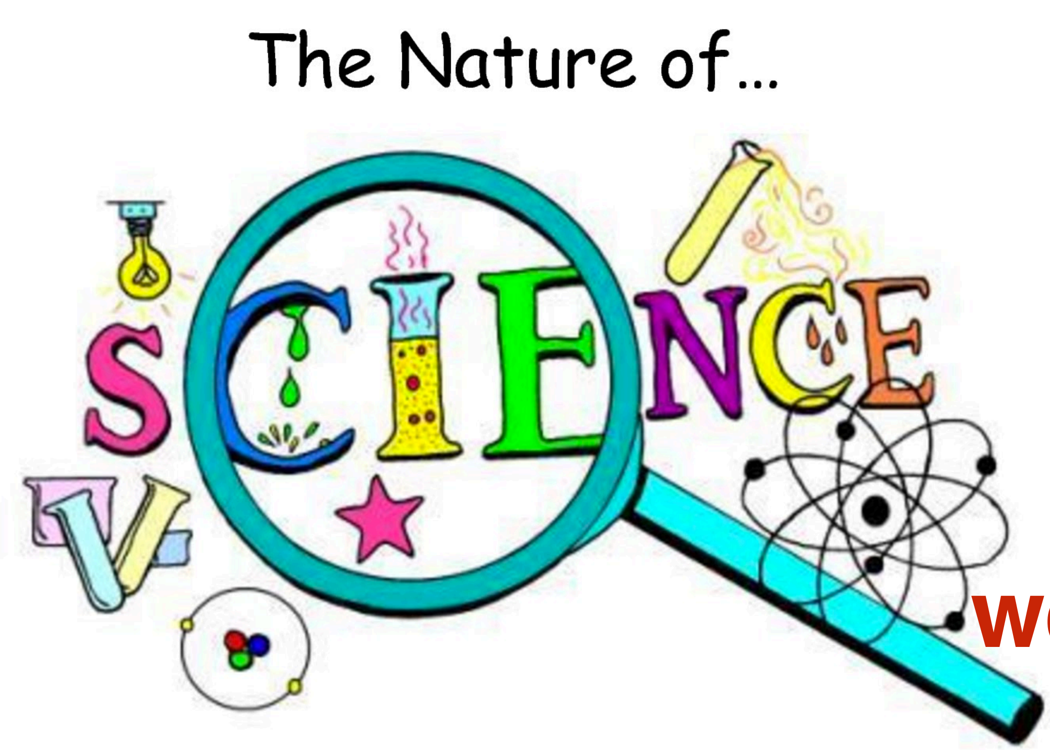
Come up with hypotheses.
Perform experiments.

Q1: What is encoded/captured in a vector?

Q2: what kinds of linguistic structures
can be captured by an RNN?

Q3: how did a given model reach a decision?
how is the architecture capturing the phenomena?

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



Treat the representations / model as an "organism".

Come up with hypotheses.
Perform experiments.

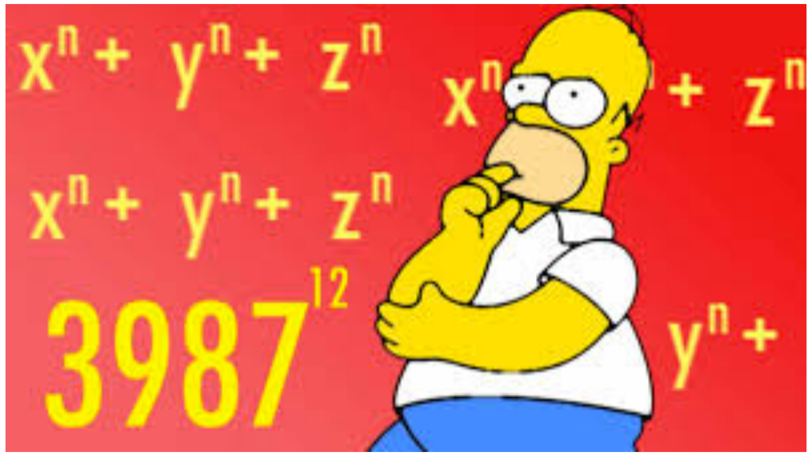
we never learned to do this in CS :(

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.



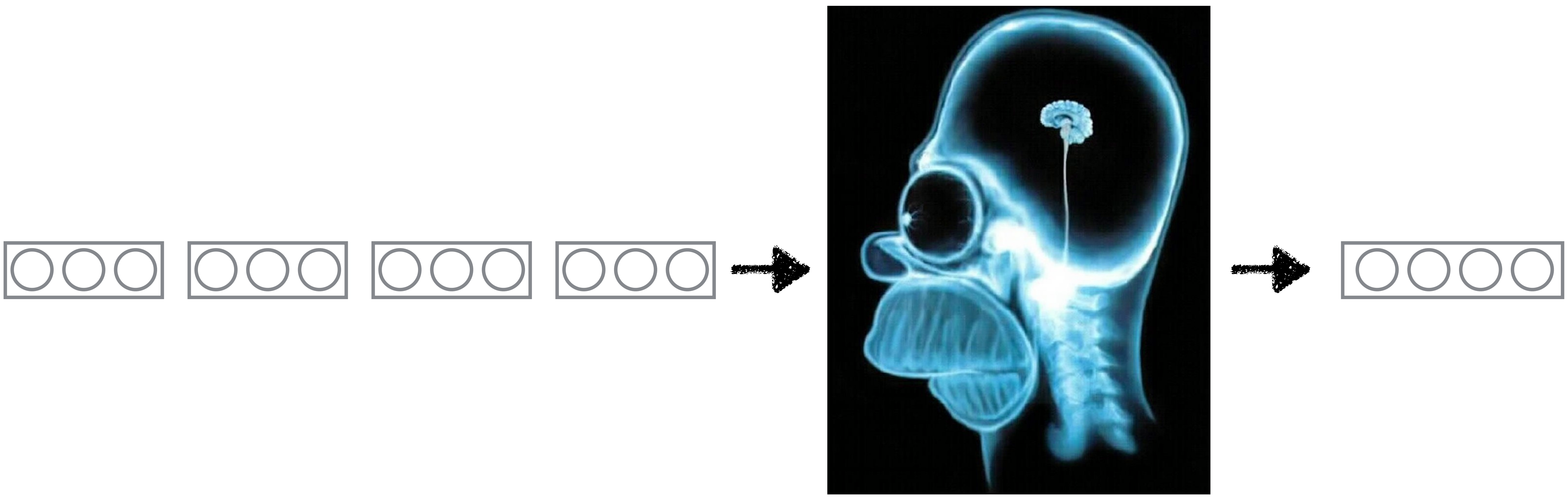
Back to a "familiar territory". Computer science. Math.

Agenda

- Formal expressive power of RNNs
- Extracting FSAs from RNNs

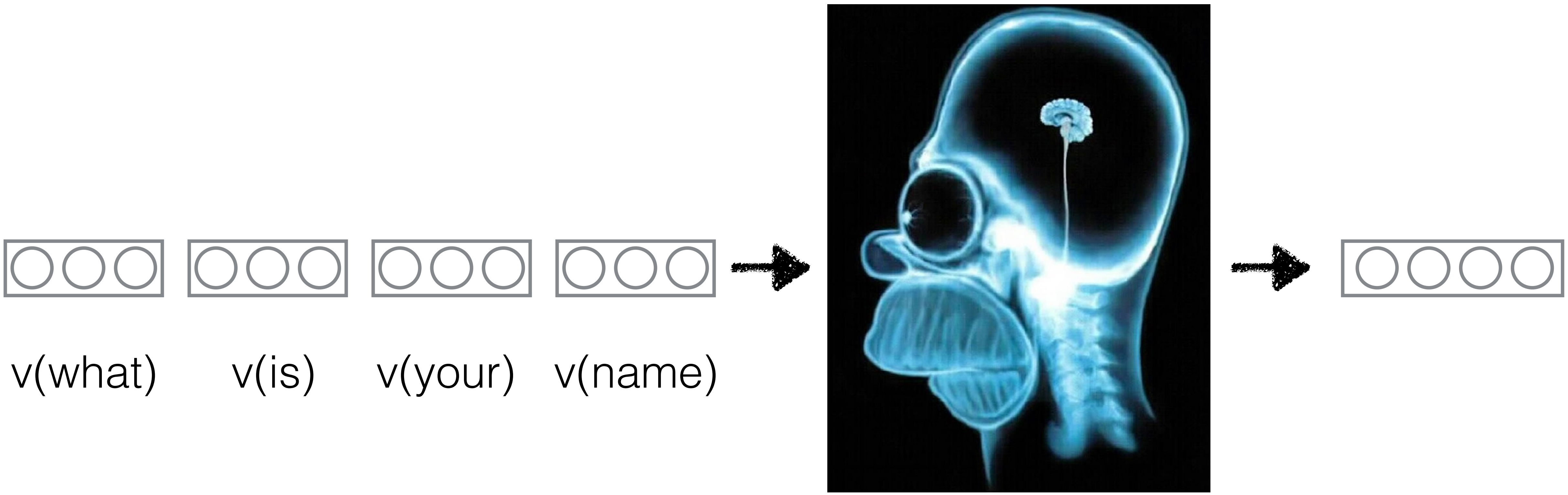
brief recap of RNNs

Recurrent Neural Networks



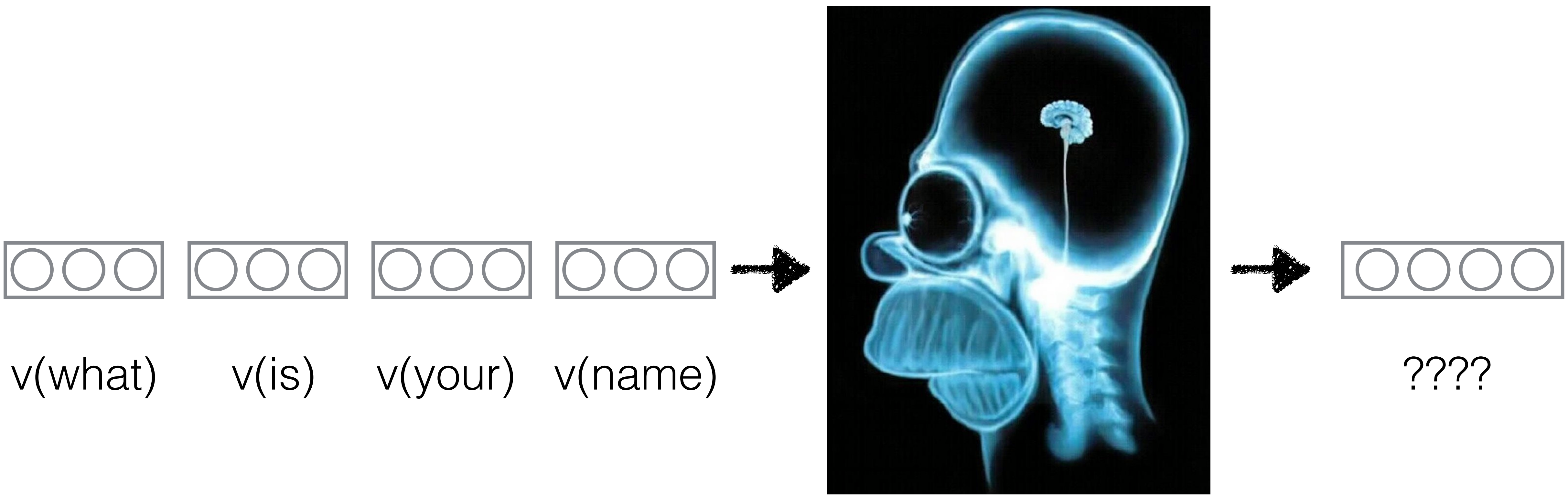
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

Recurrent Neural Networks



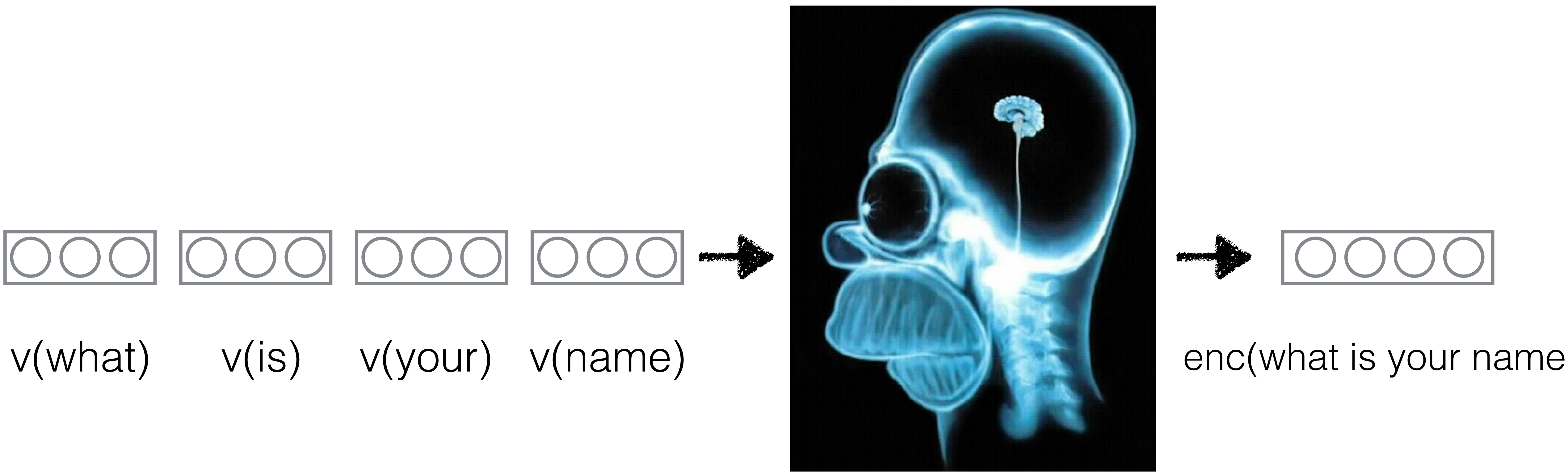
- Very strong models of sequential data.
- Function from *n* vectors to a single vector.

Recurrent Neural Networks



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

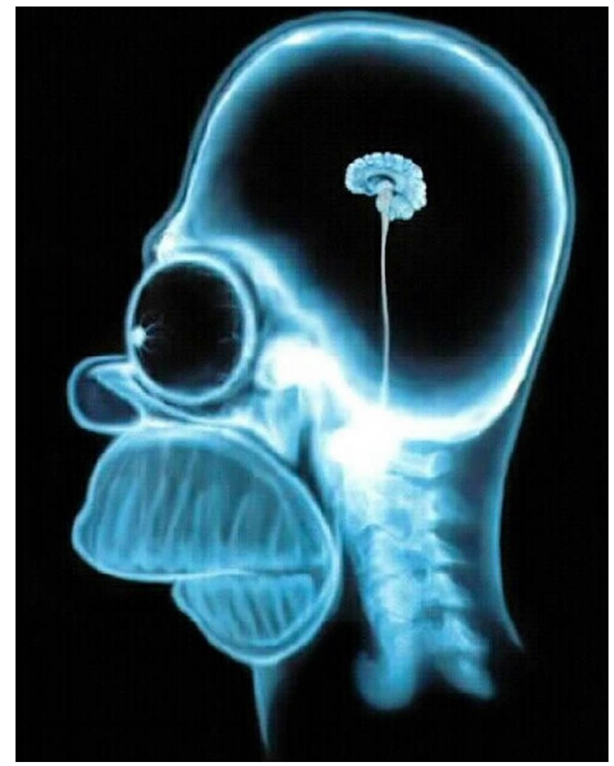
Recurrent Neural Networks



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

Recurrent Neural Networks

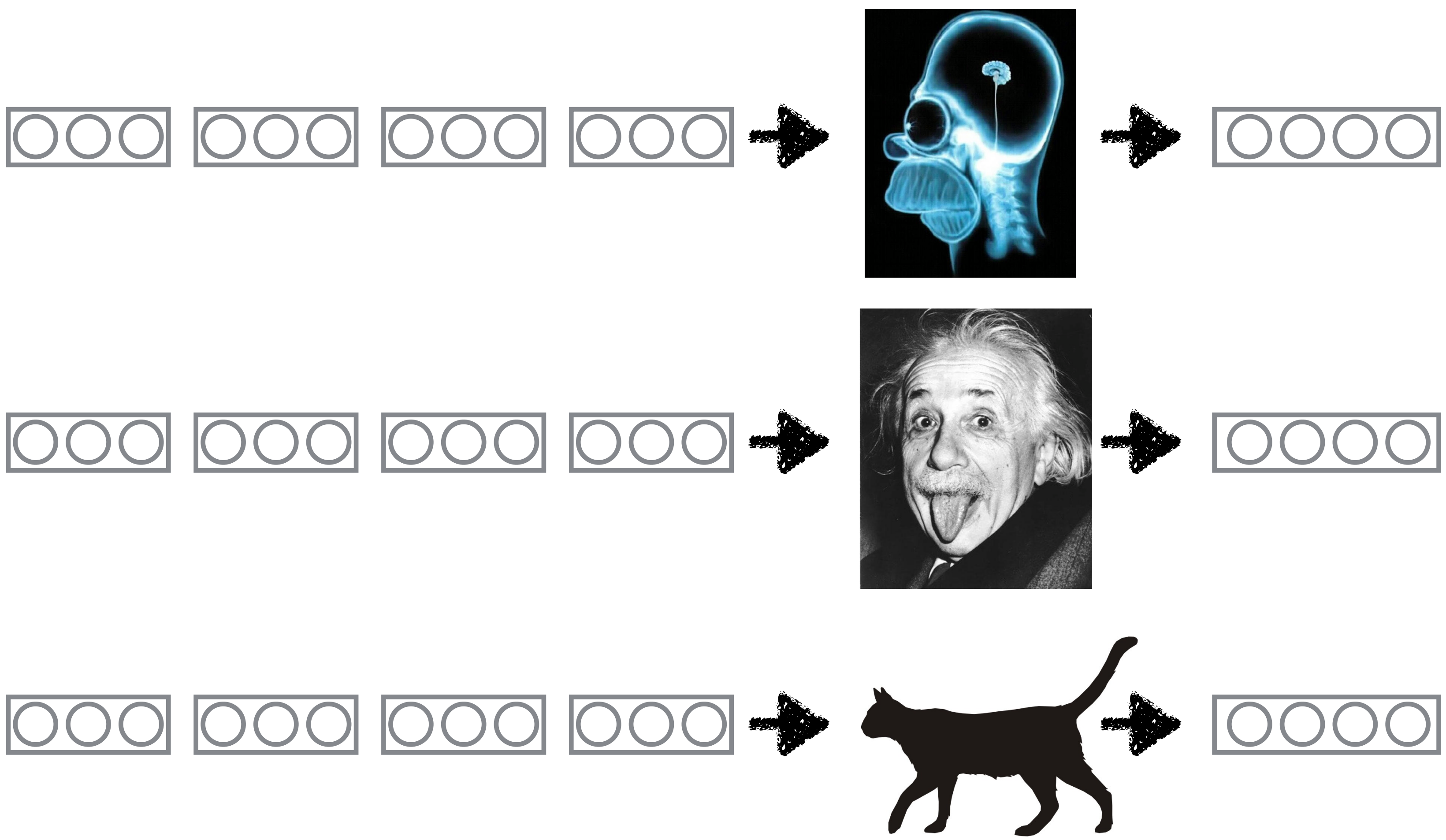
v(what) v(is) v(your) v(name)



enc(what is your name)

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

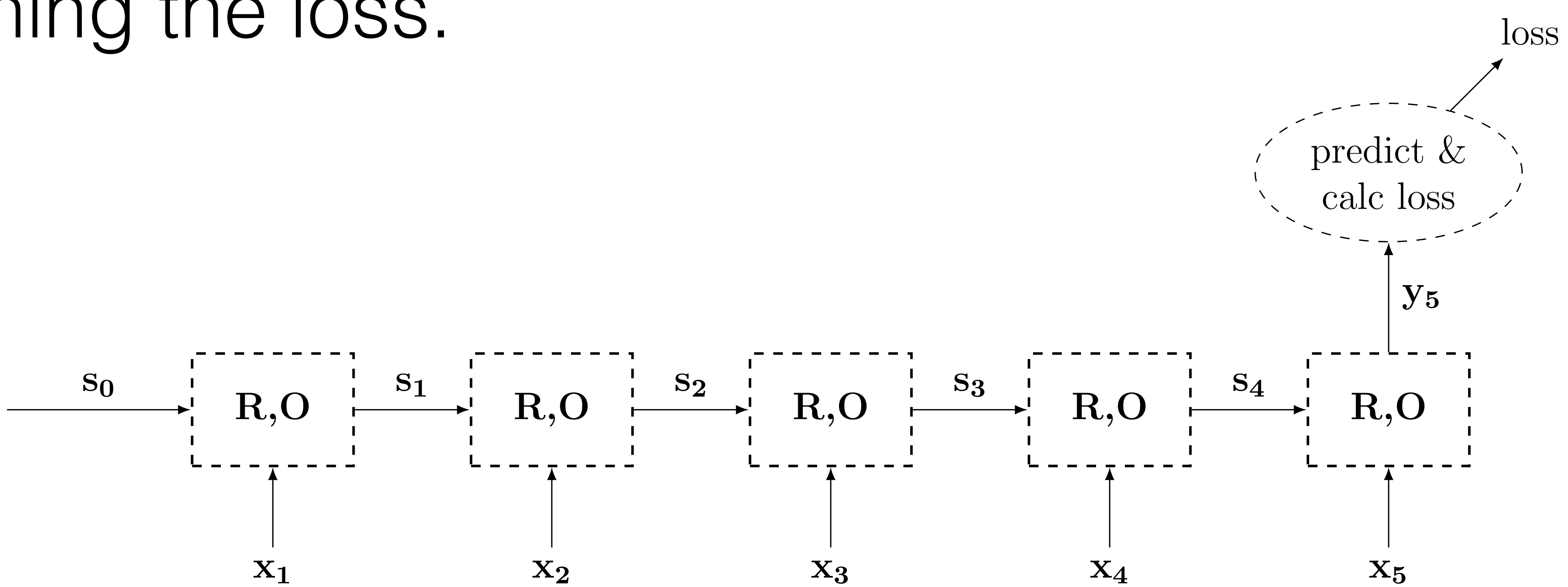
Recurrent Neural Networks



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

Recurrent Neural Networks

Defining the loss.



Acceptor: Read in a sequence. Predict from the end state.
Backprop the error all the way back.
Train the network to capture meaningful information

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

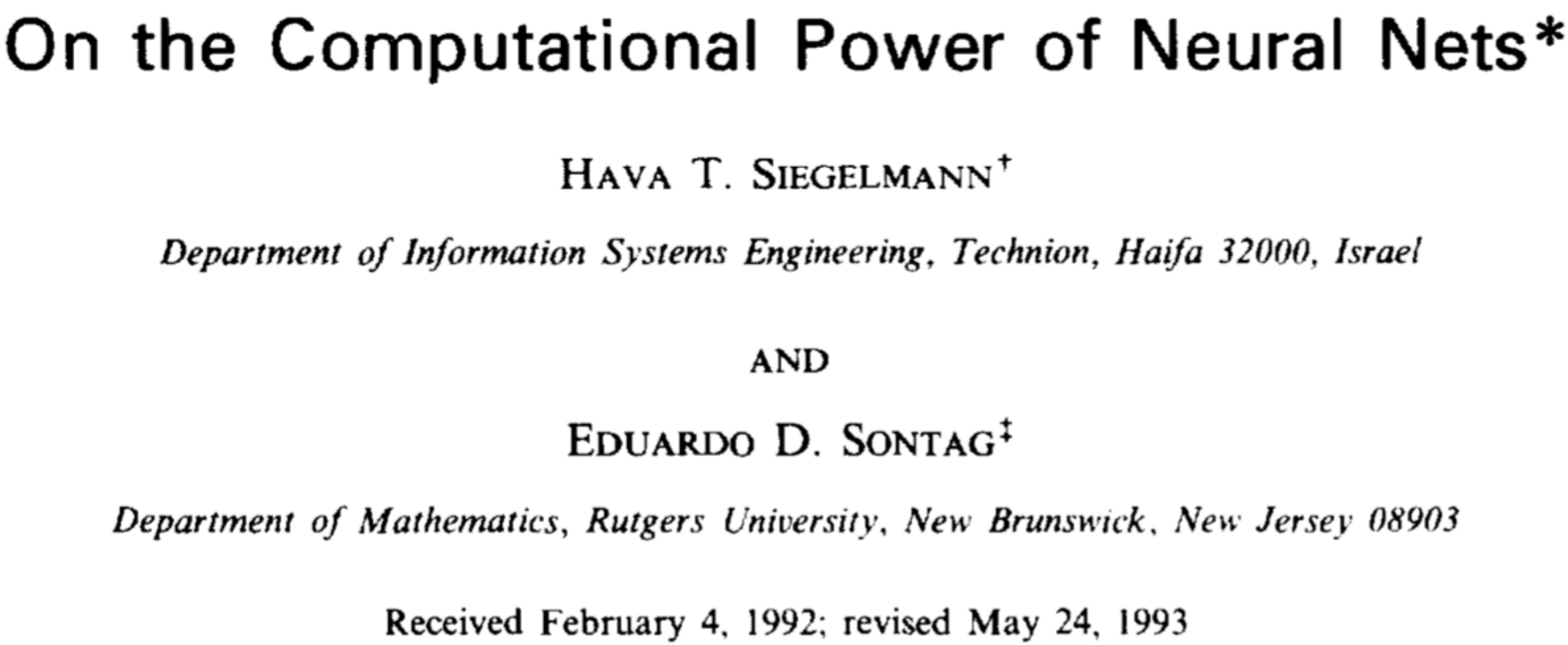
EDUARDO D. SONTAG‡

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*,†

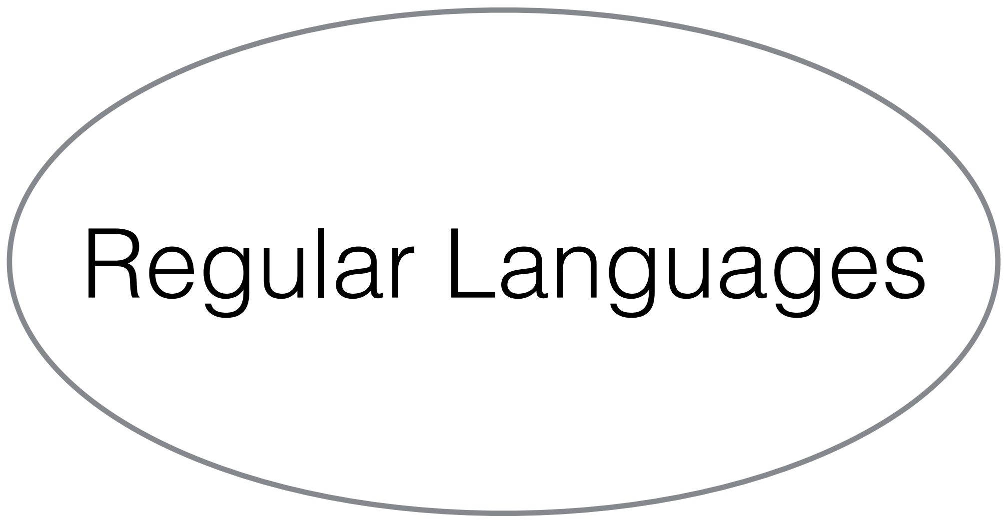
counter machines are
Finite State Automata with k counters.

INC, DEC, Compare0

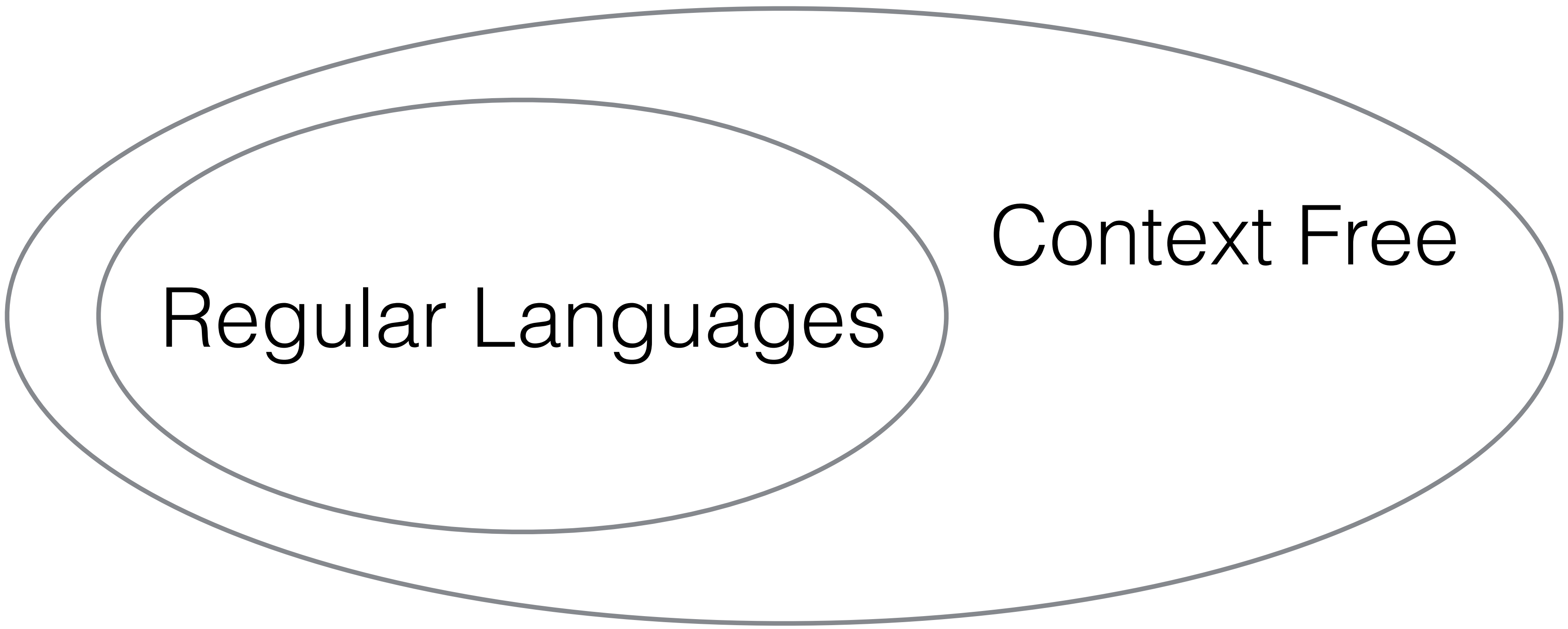
Yorktown Heights, New York

(1968)

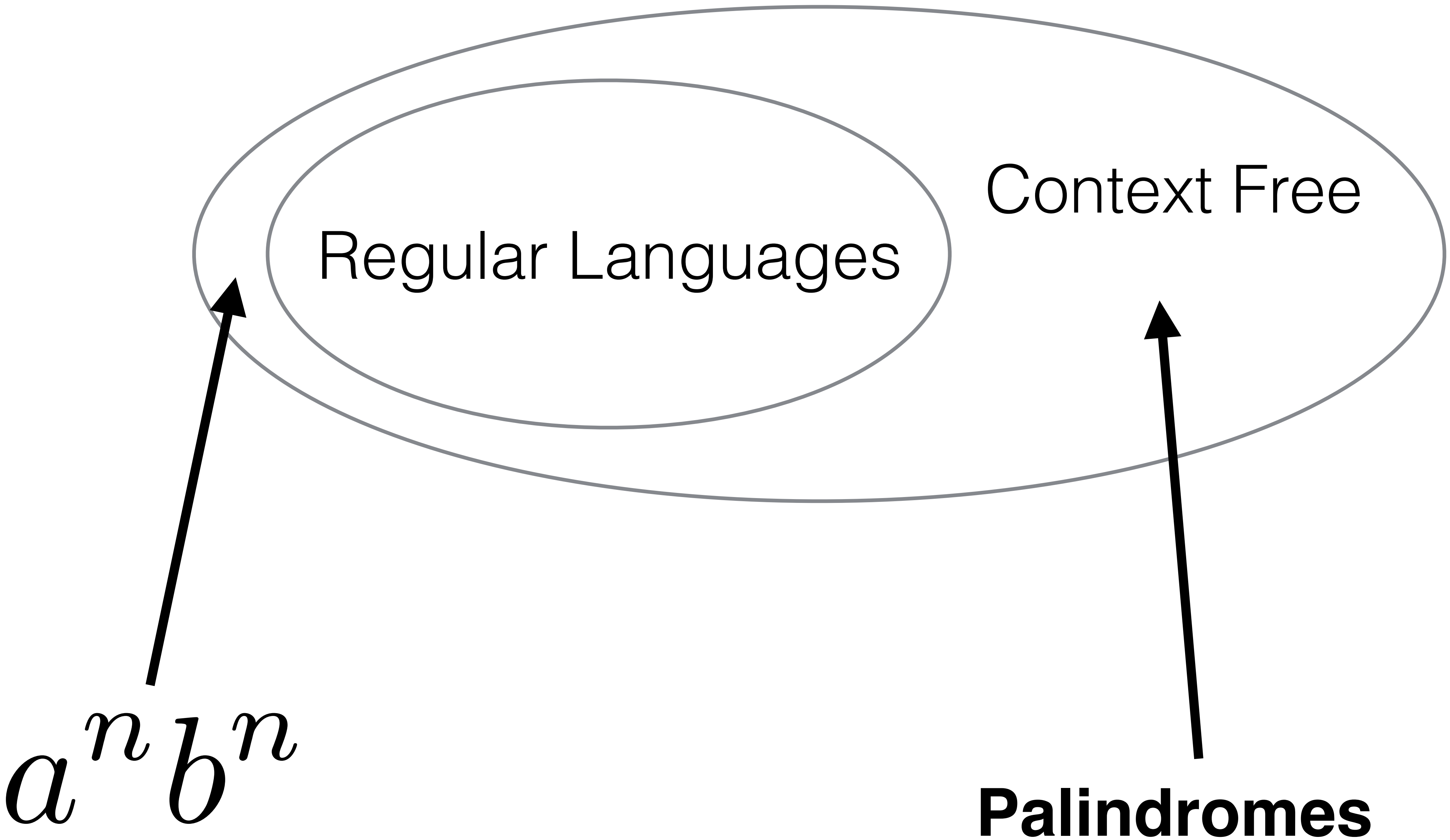
Chomsky Hierarchy



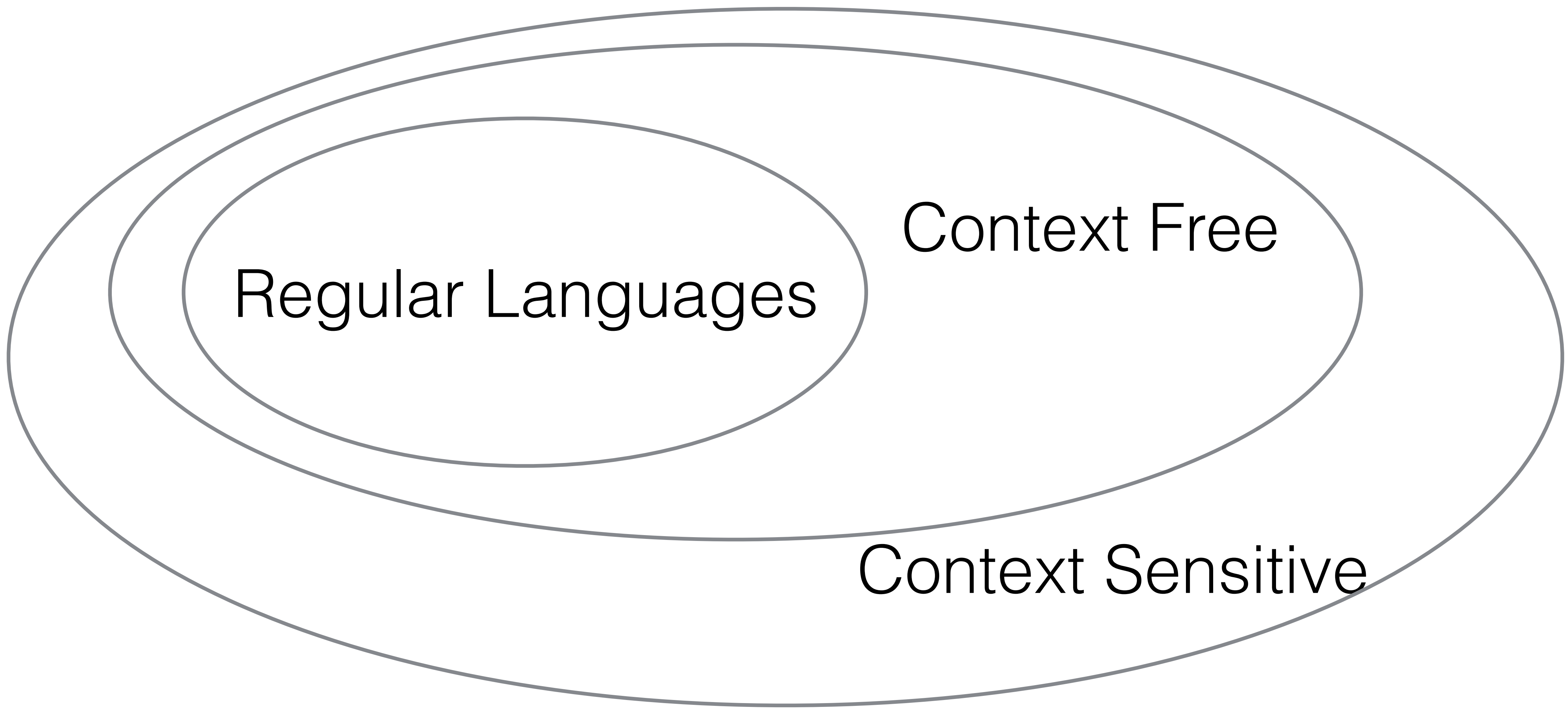
Chomsky Hierarchy



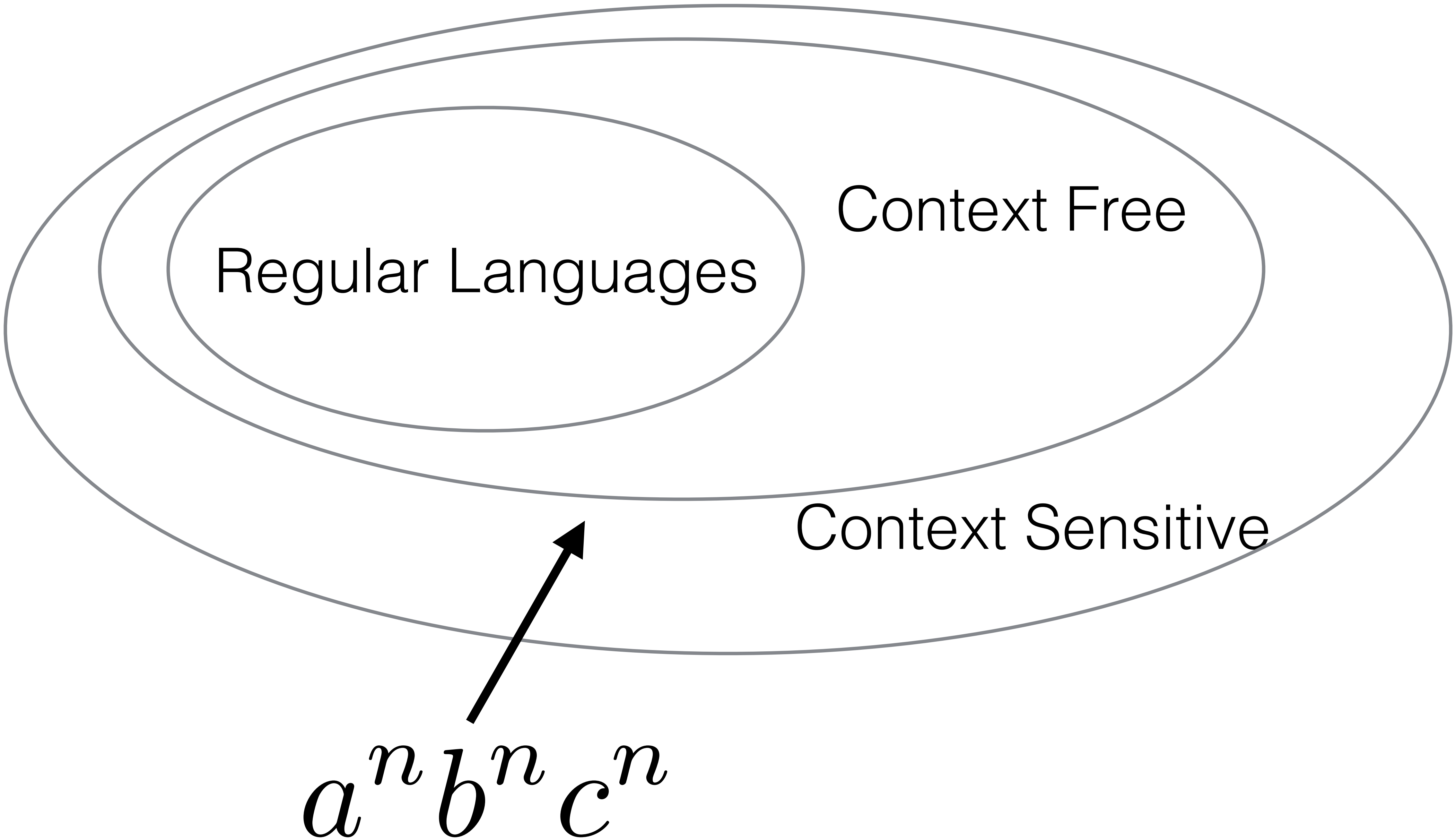
Chomsky Hierarchy



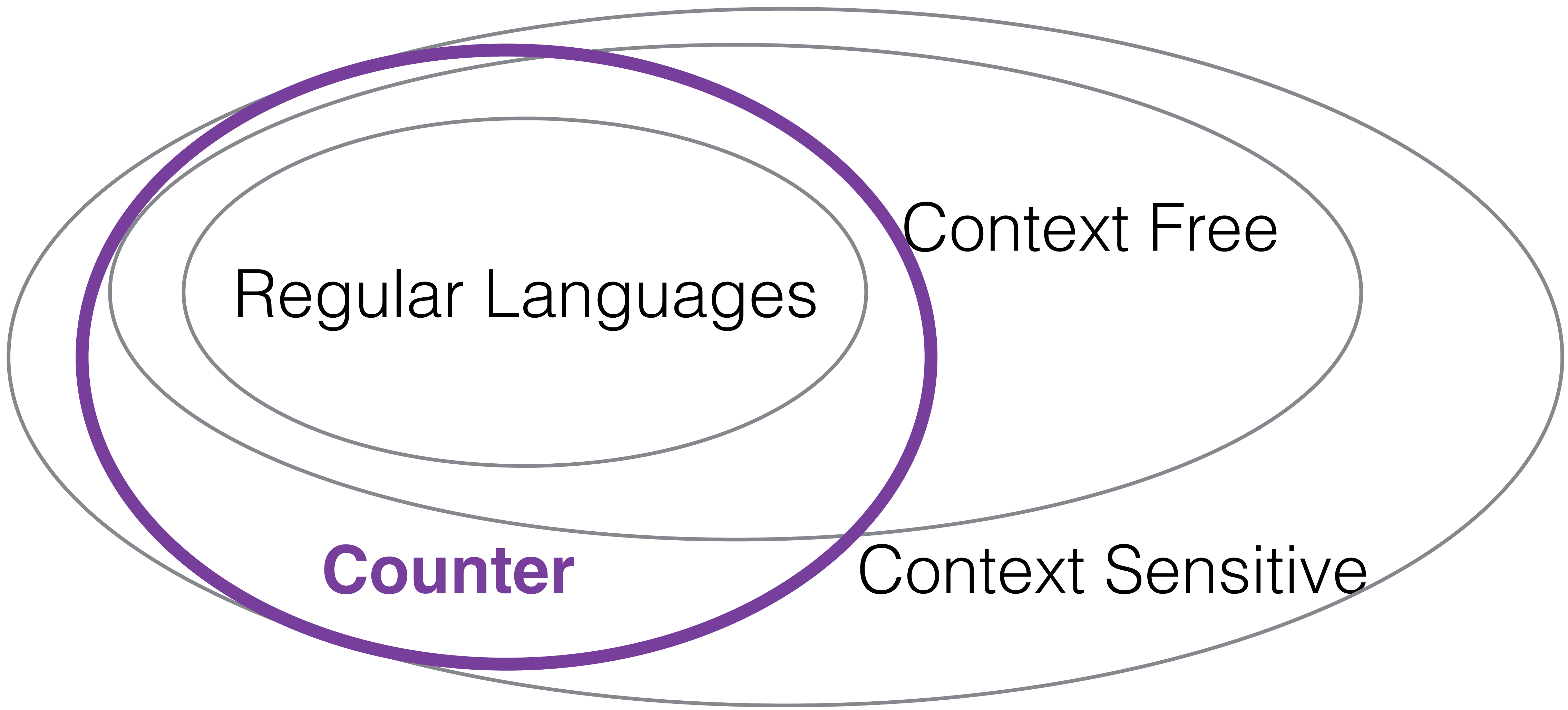
Chomsky Hierarchy



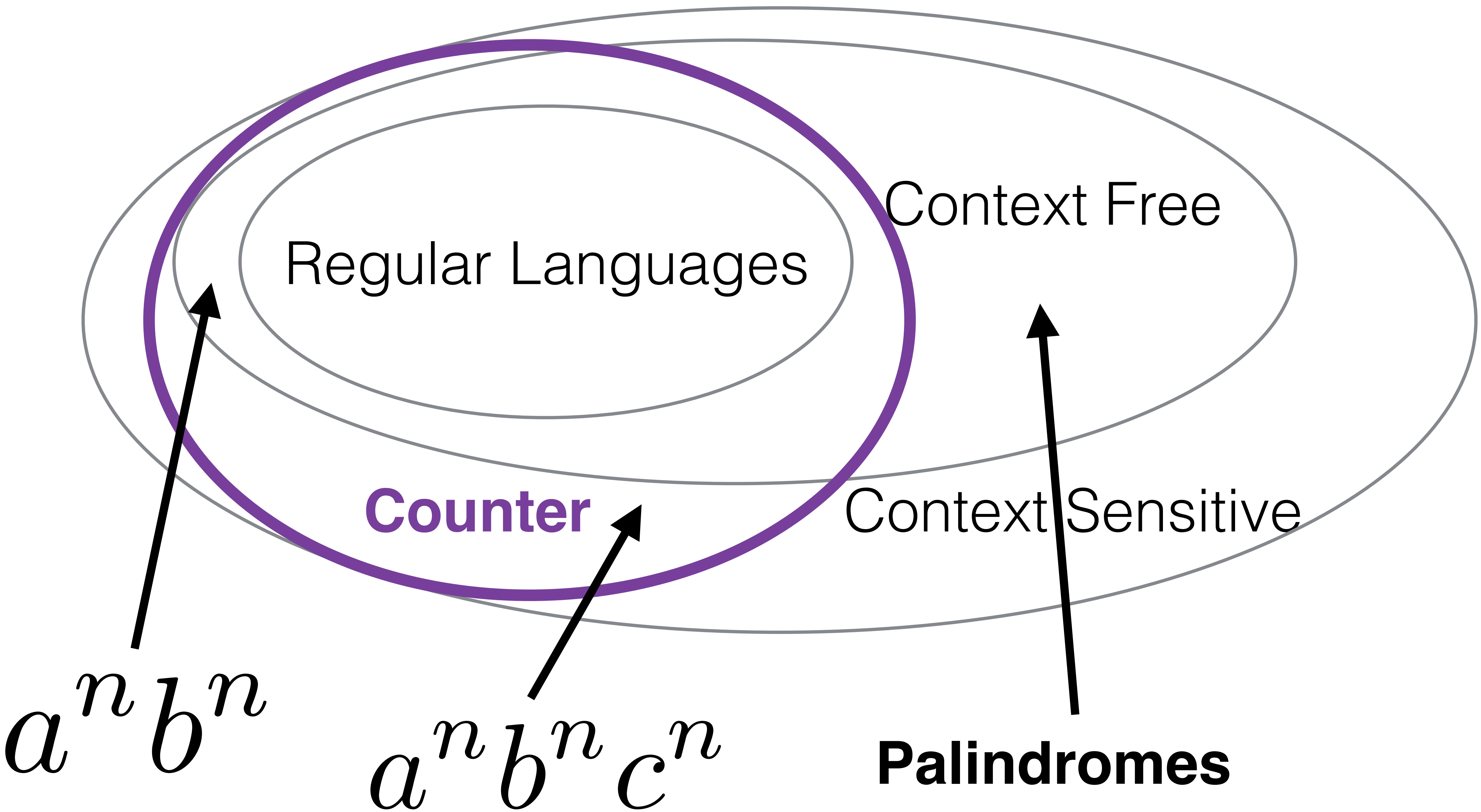
Chomsky Hierarchy



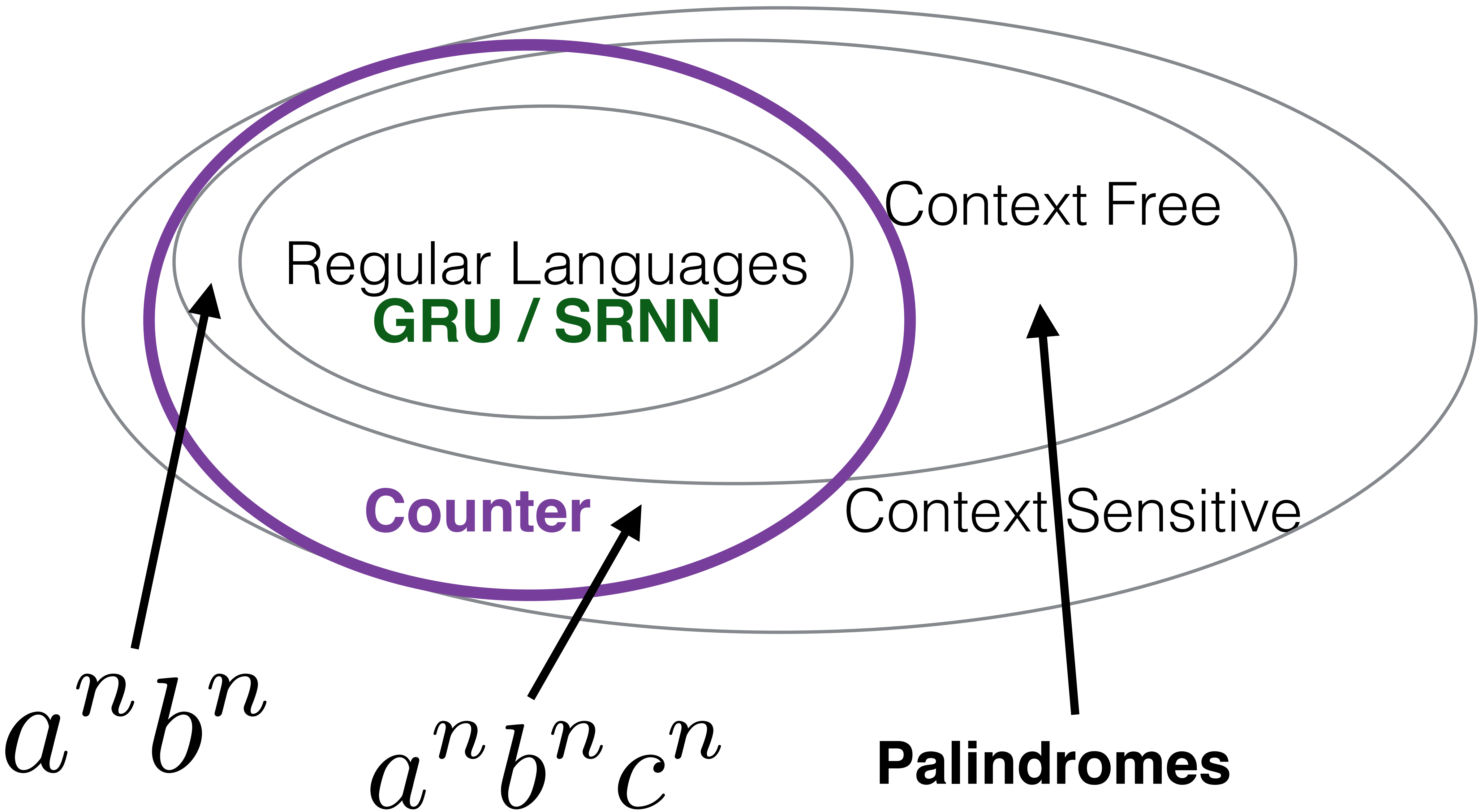
Power of Counting



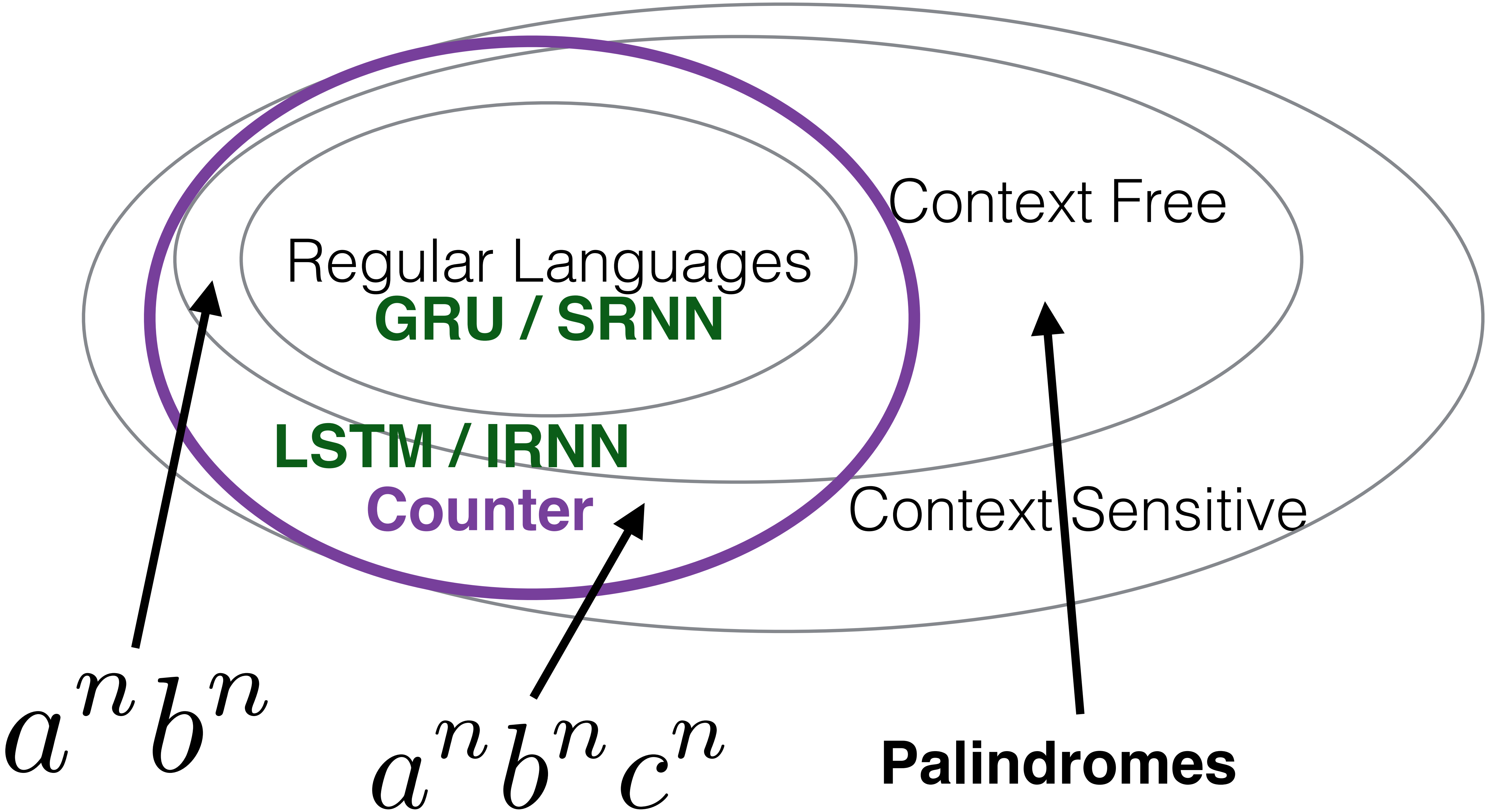
Power of Counting



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)

compare to zero is easy

-1 , 1
(via tanh)

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)

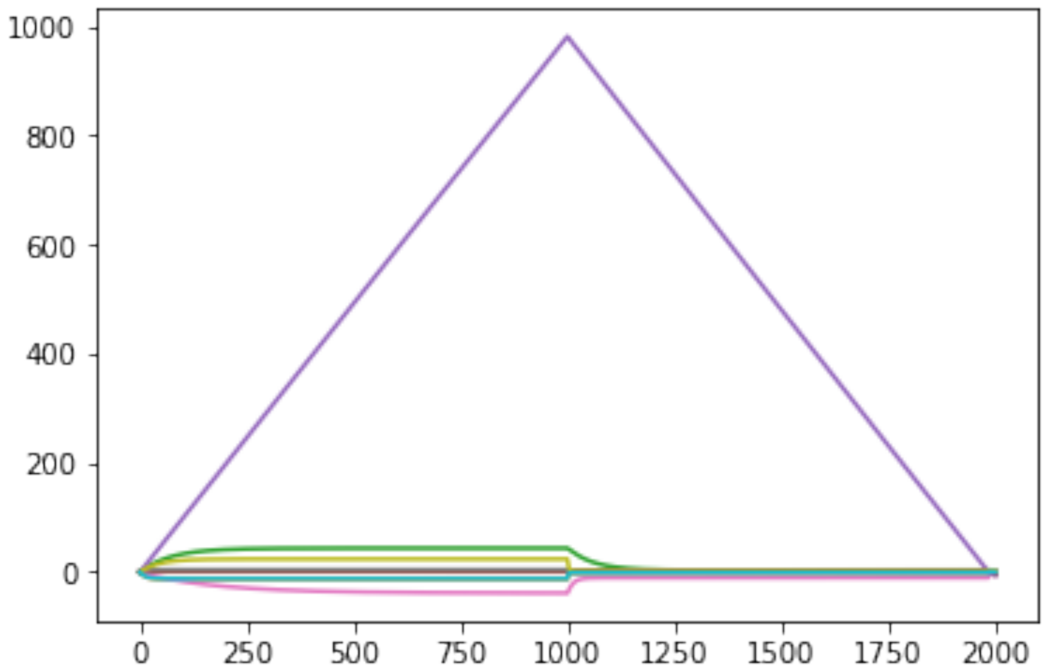
Counting in some other way?

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

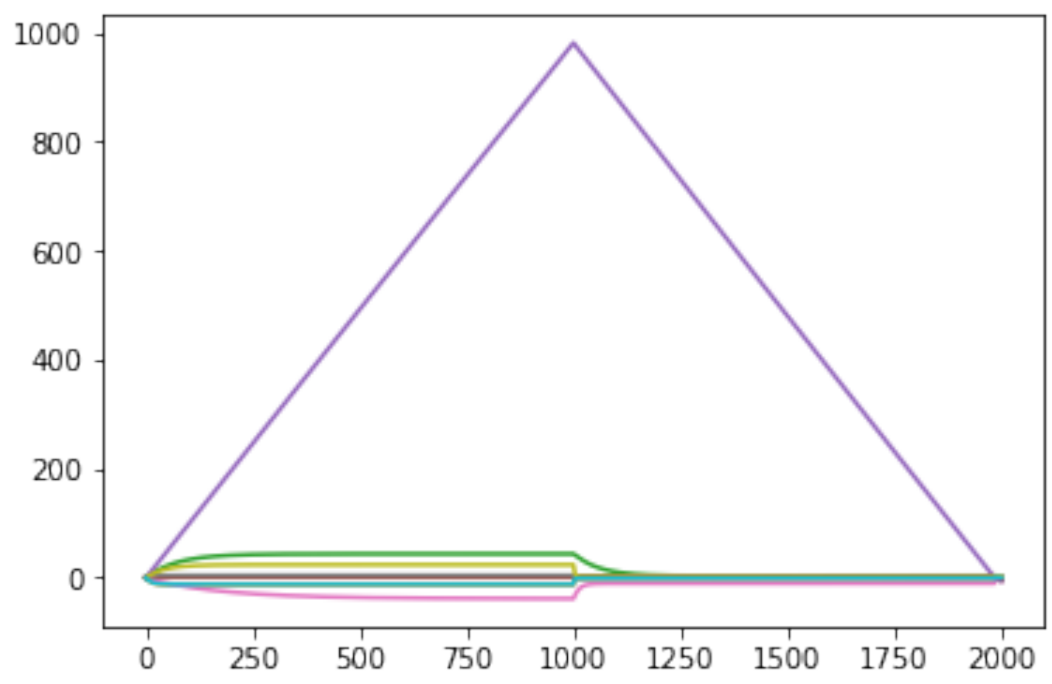
LSTM vs. GRU



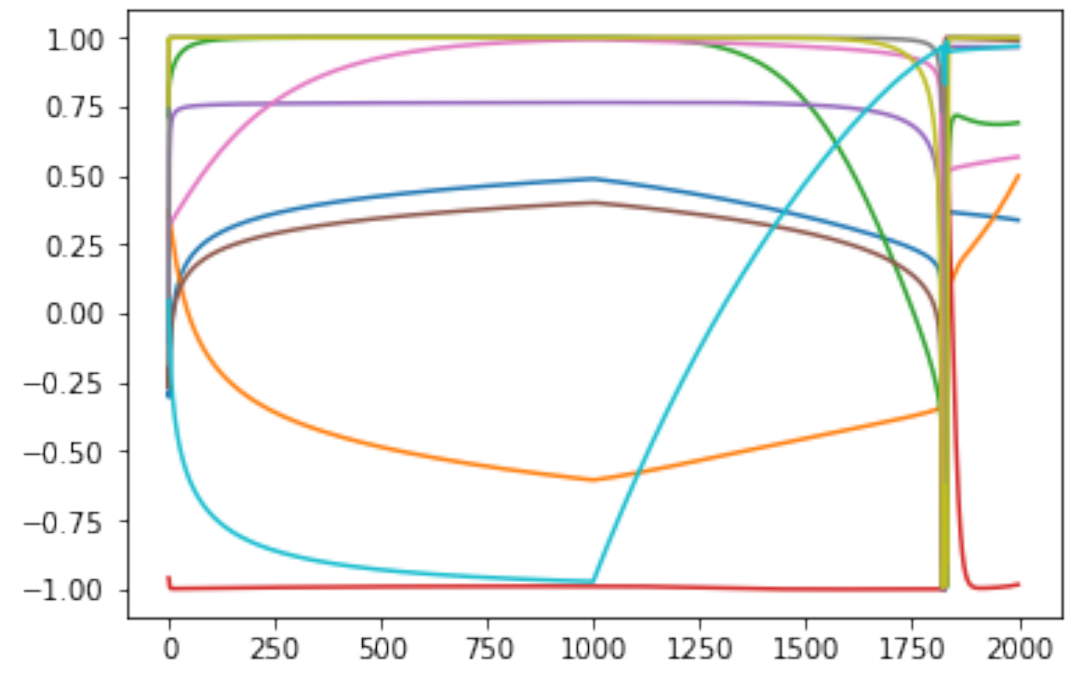
(a) $a^n b^n$-LSTM on $a^{1000}b^{1000}$

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

LSTM vs. GRU



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

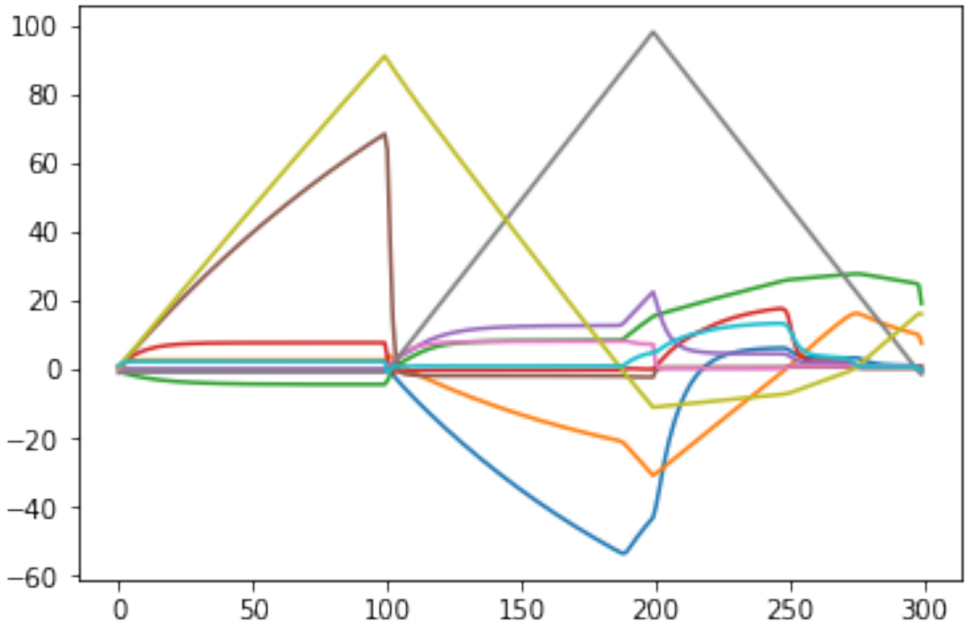


(c) $a^n b^n$-GRU on $a^{1000}b^{1000}$

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

GRU starts to fail at n=38

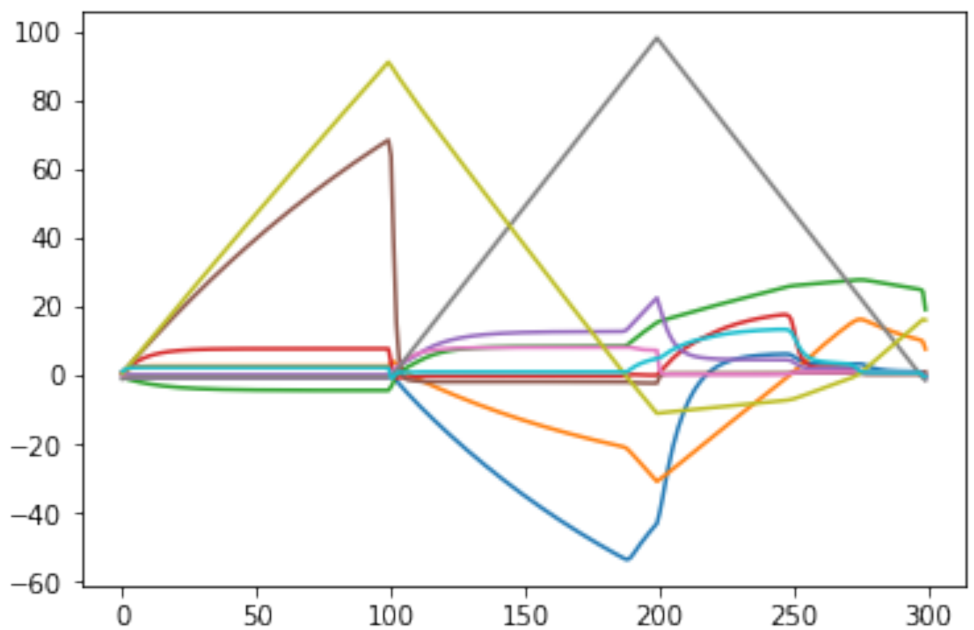
LSTM vs. GRU



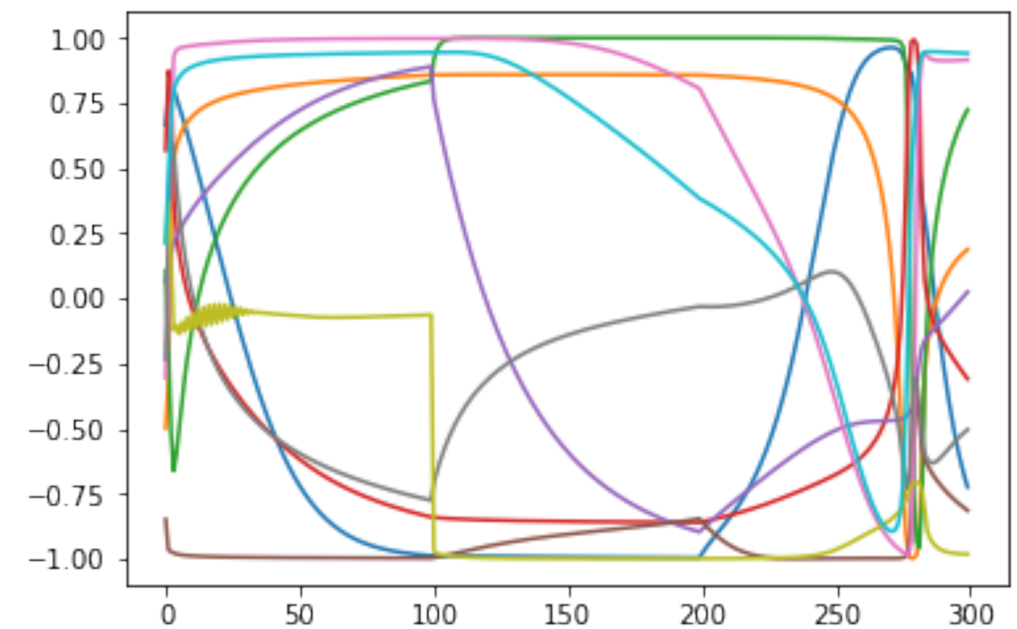
(b) $a^n b^n c^n$-LSTM on $a^{100} b^{100} c^{100}$

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

BIU NLP

TECHNION Israel Institute of Technology

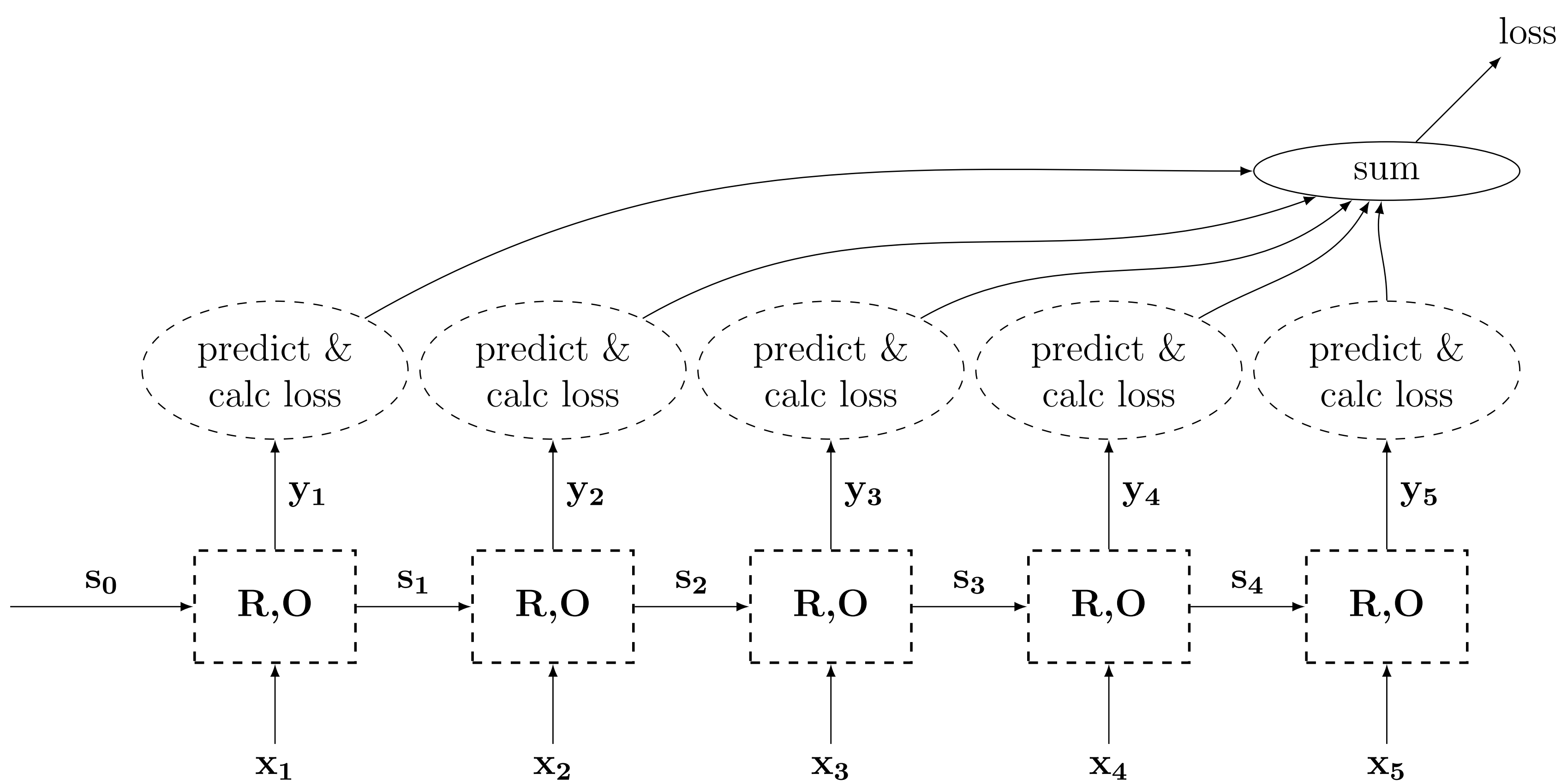
Extracting Automata from Recurrent Neural Networks Using Queries and Counterexamples

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

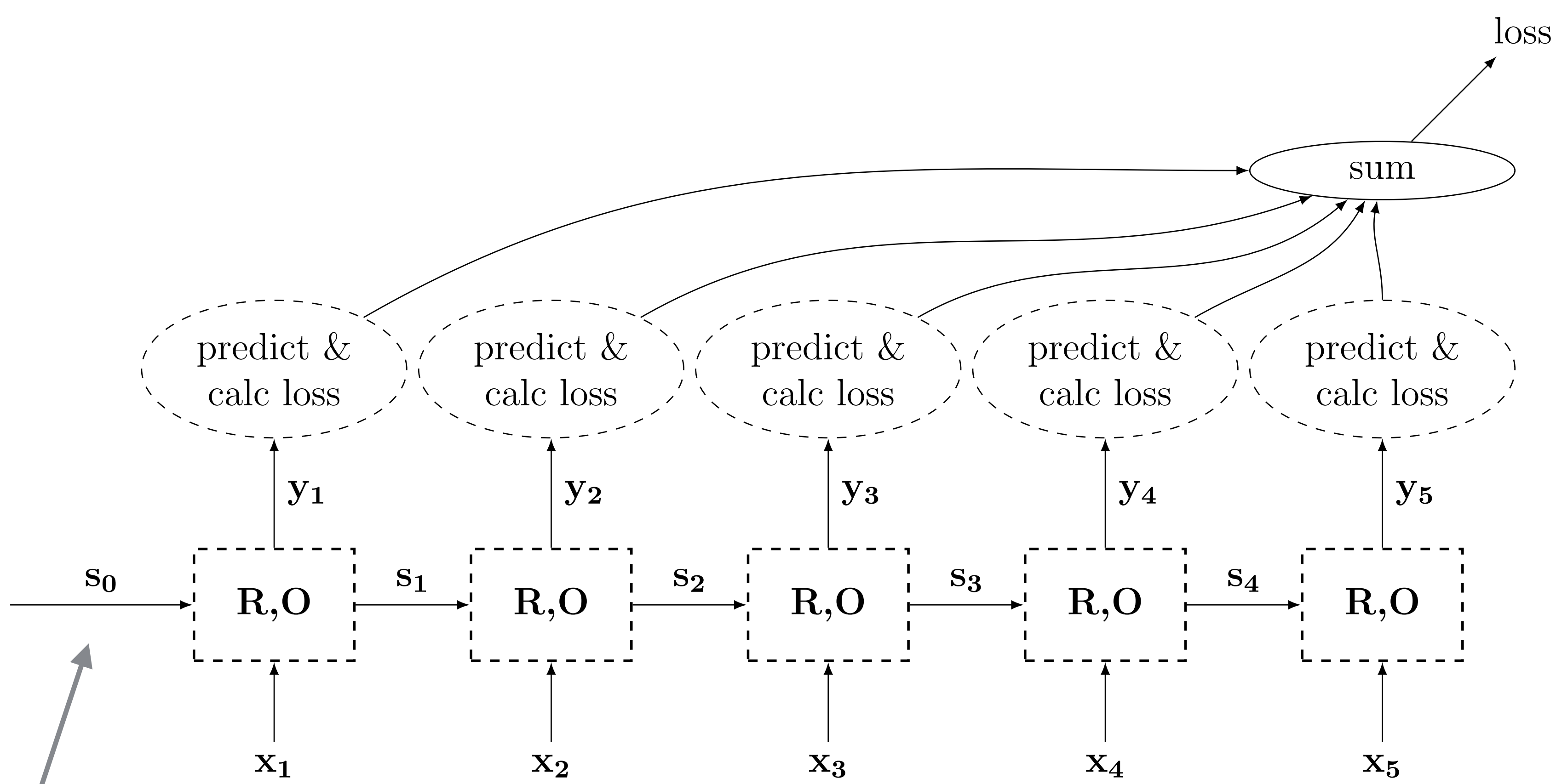
(ICML 2018)

RNN acceptors as State Machines

RNN acceptors as State Machines

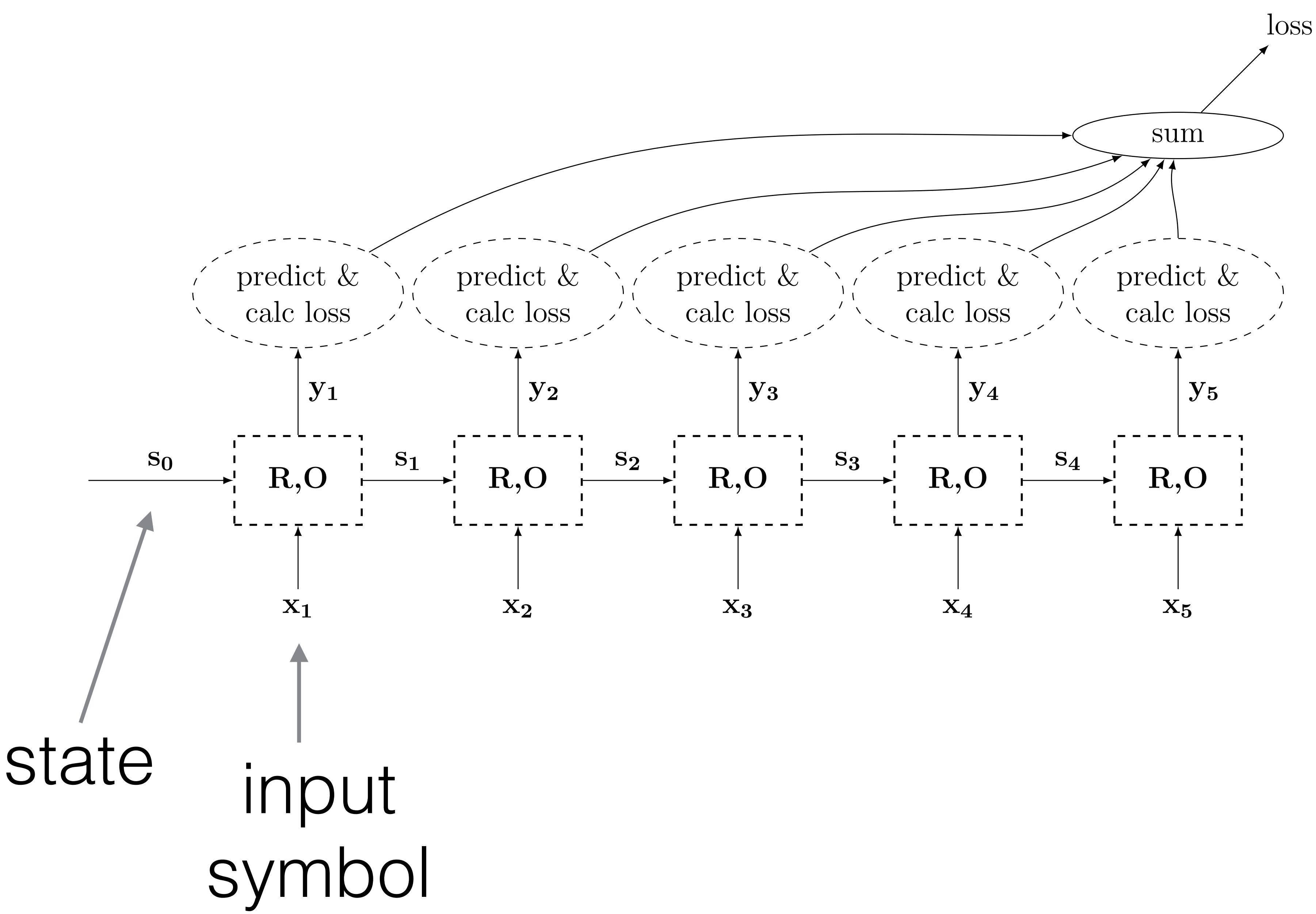


RNN acceptors as State Machines

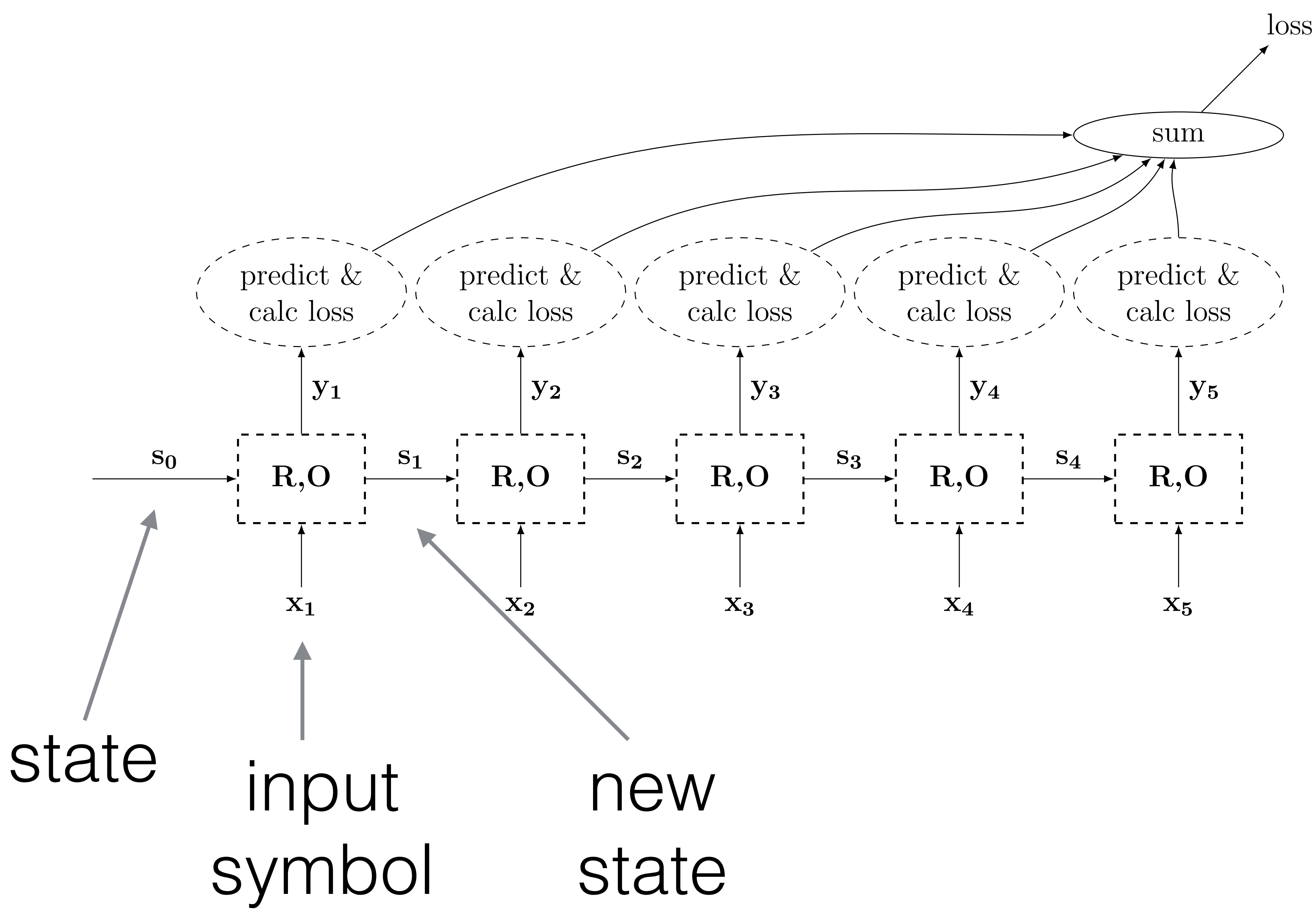


state

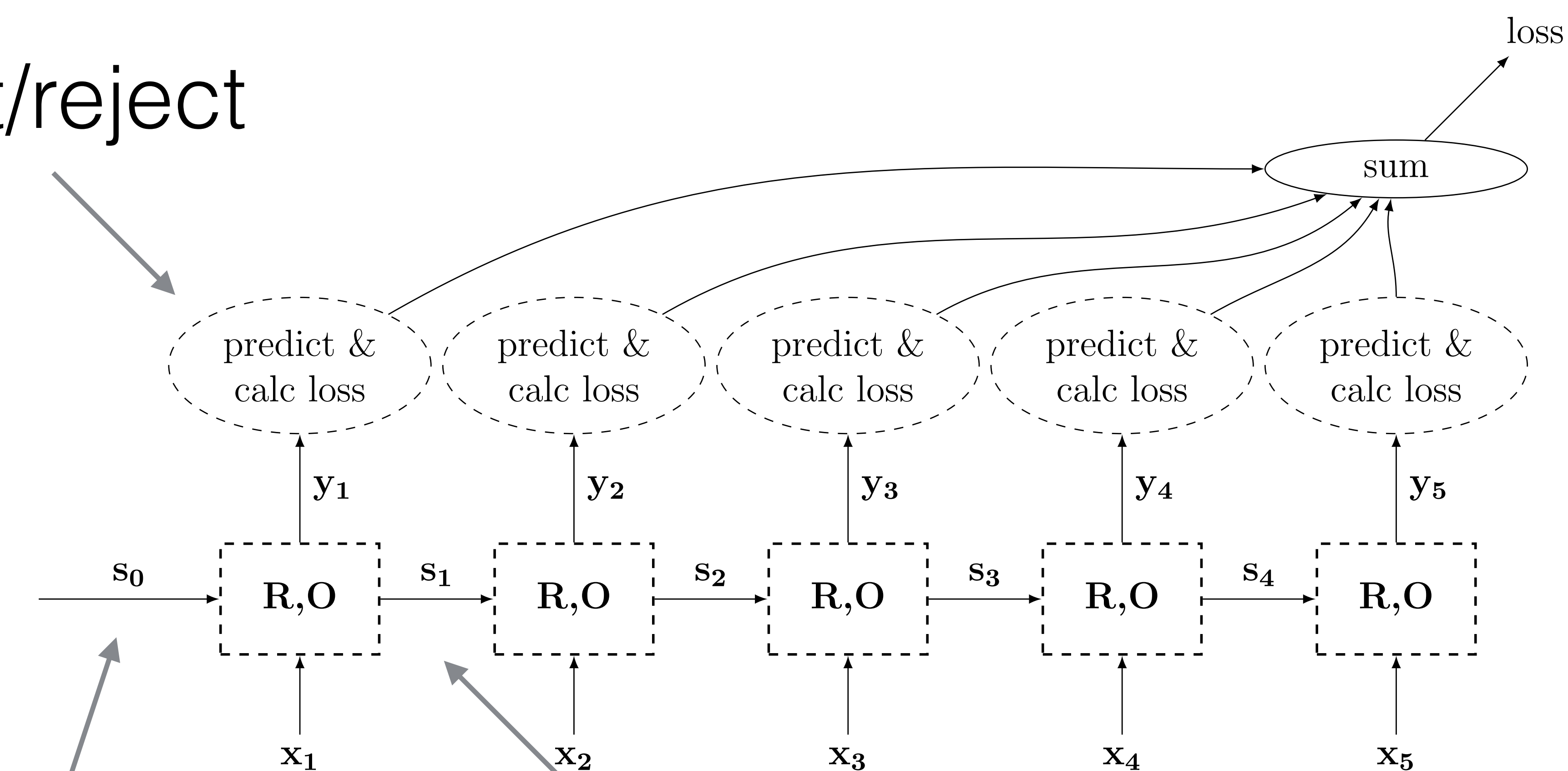
RNN acceptors as State Machines



RNN acceptors as State Machines



RNN acceptors as State Machines



accept/reject

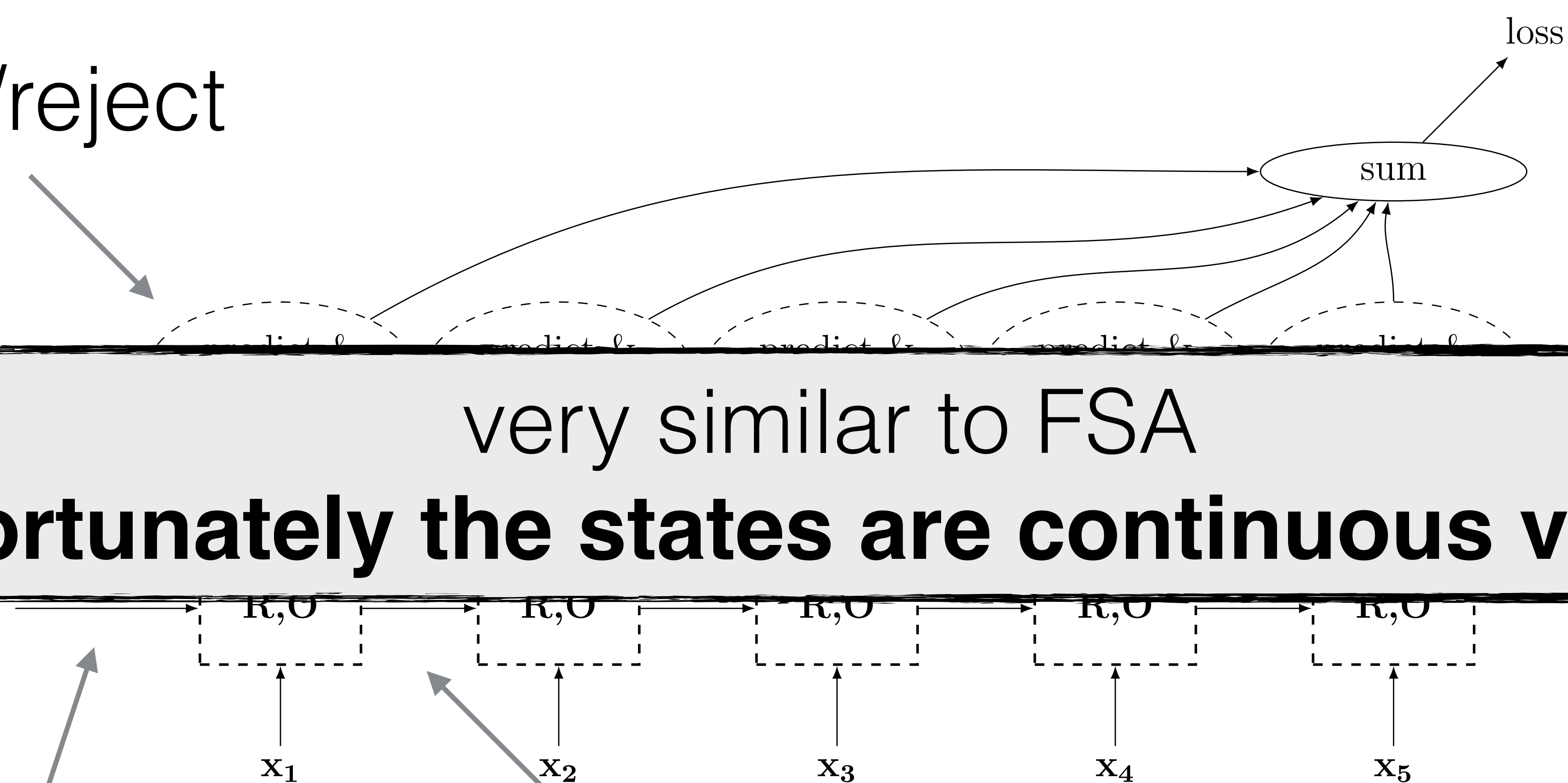
state

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

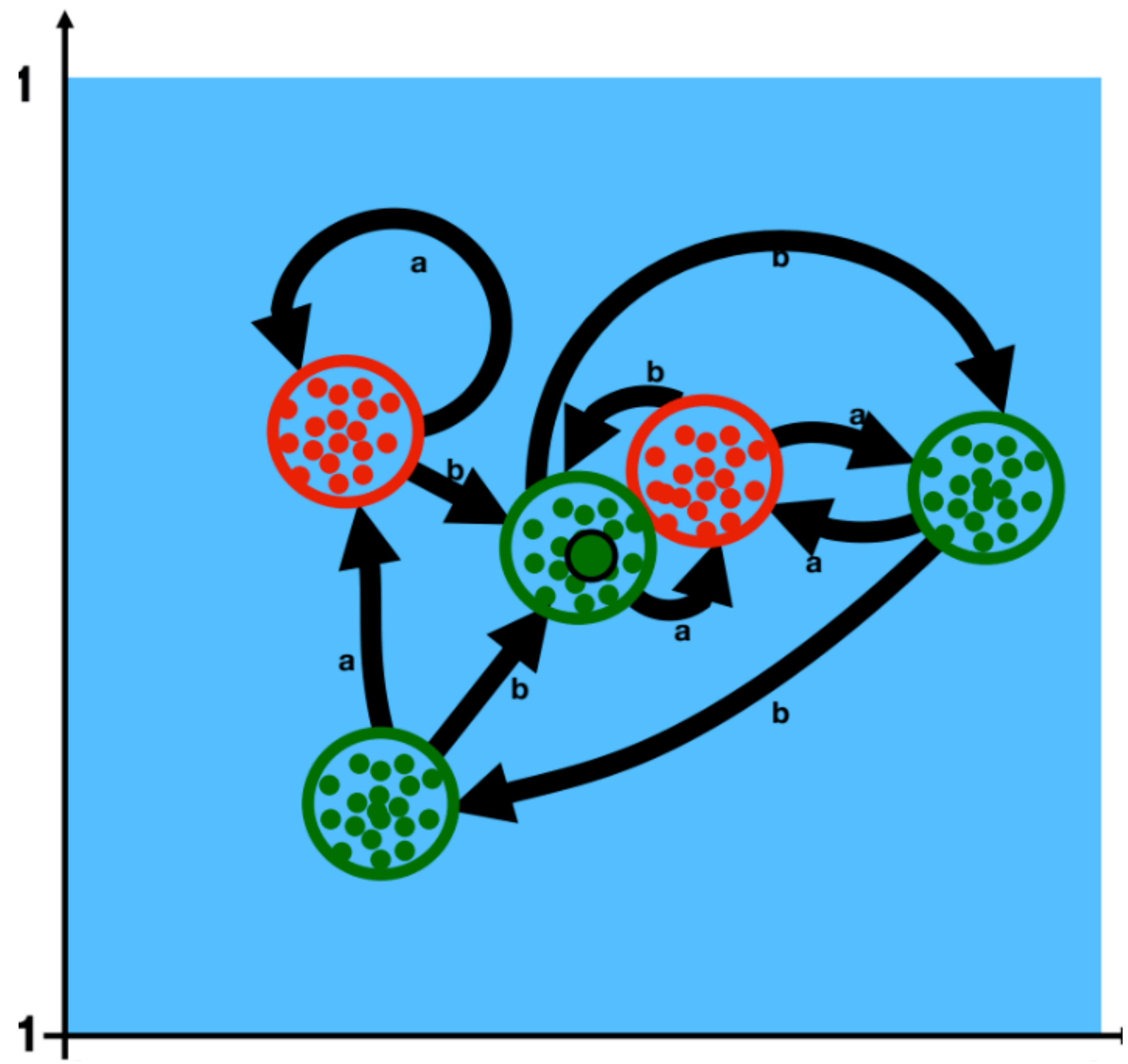
Easy. Just run the word through the RNN.

Equivalence Queries

Hard. Requires some trickery.

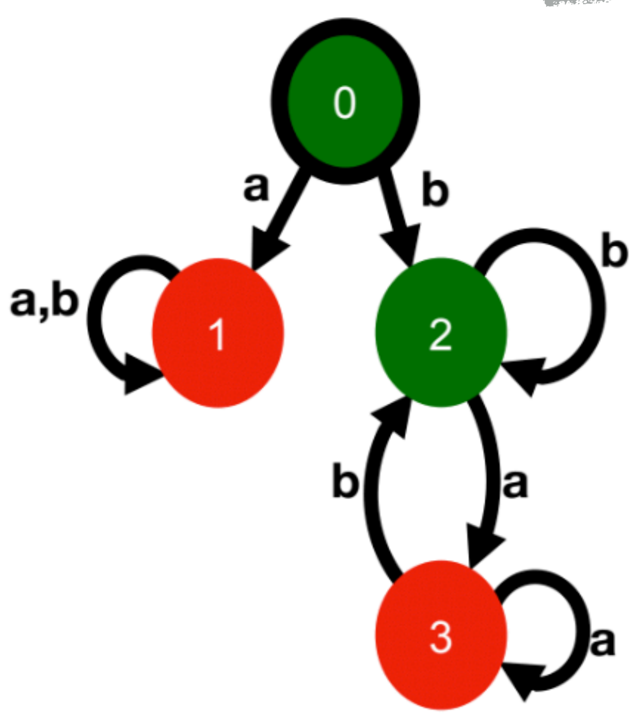
Answering Equivalence Queries

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

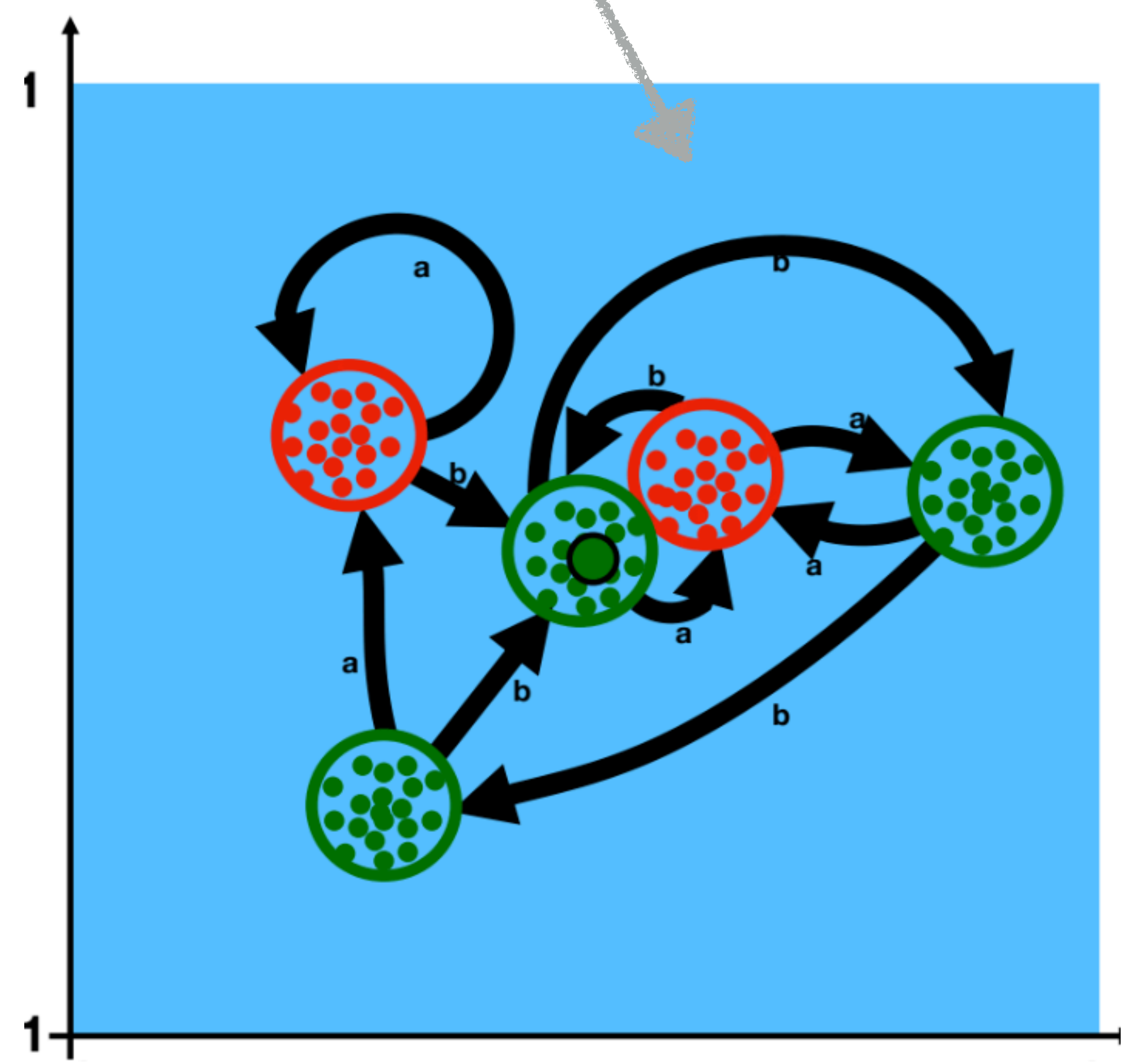


Answering Equivalence Queries

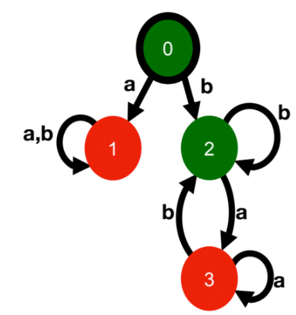
- Compare L* **Query FSA** to **RNN-Abstract-FSA**.

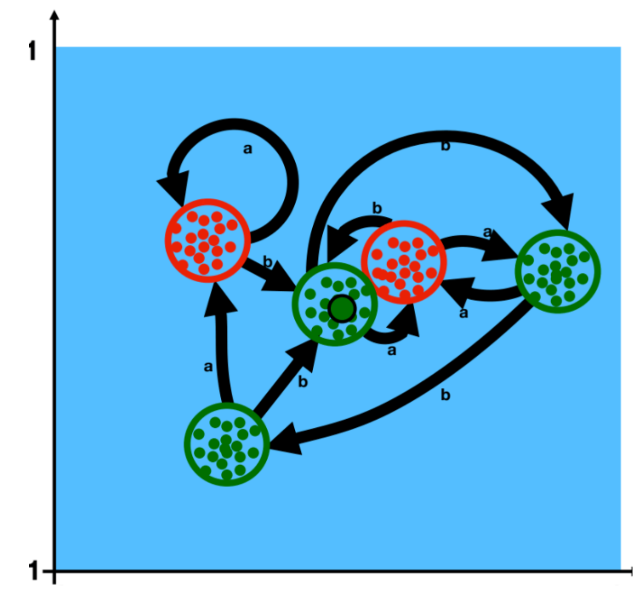


=??



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

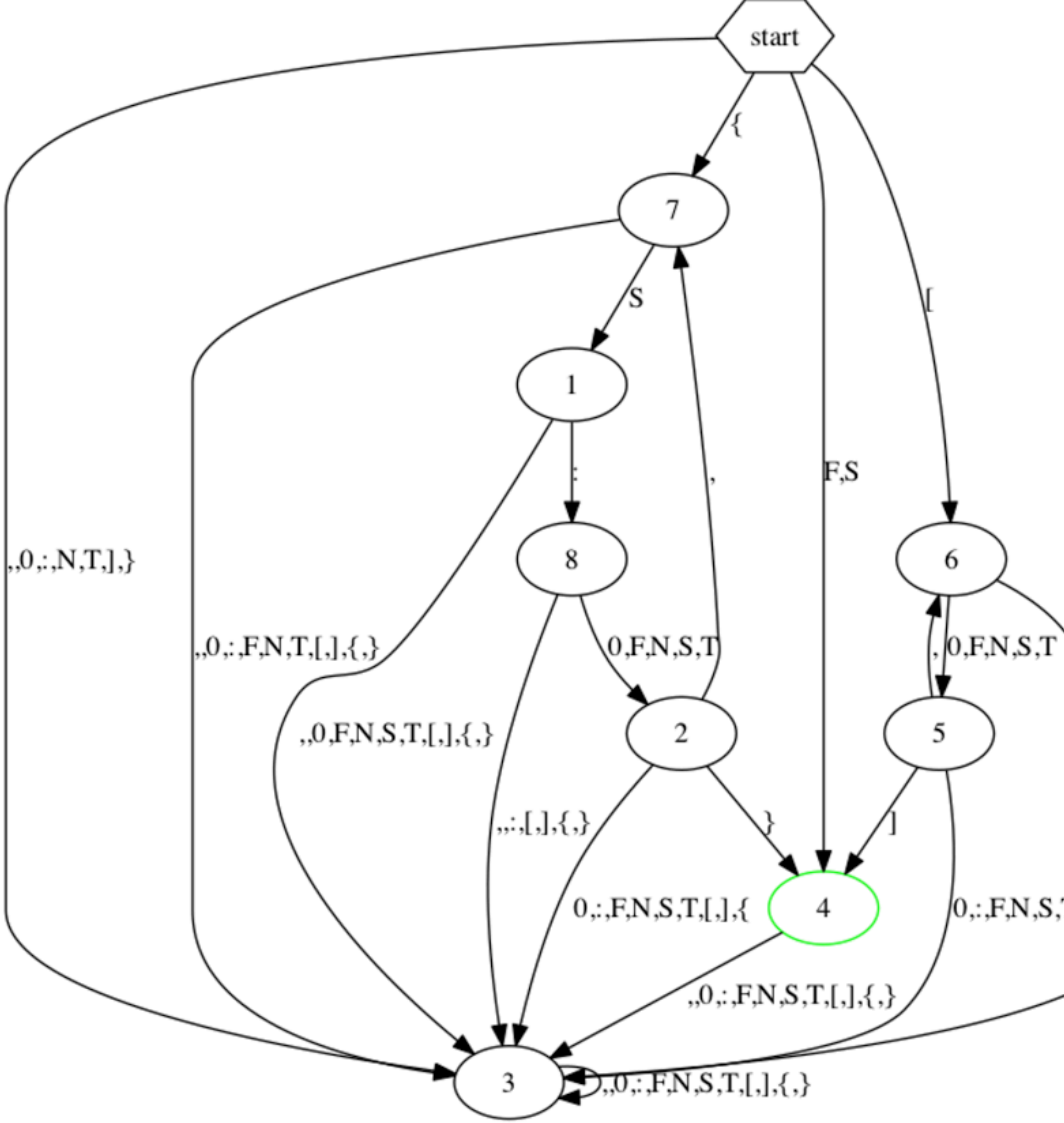
- **Many random FSAs:**
    - 5 or 10 states, alphabet sizes of 3 or 5
- LSTM/GRU with 50, 100, 500 dimensions.
- The FSAs were **learned well** by LSTM / GRU
- And **recovered well** by L*.

"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,[,],{,}

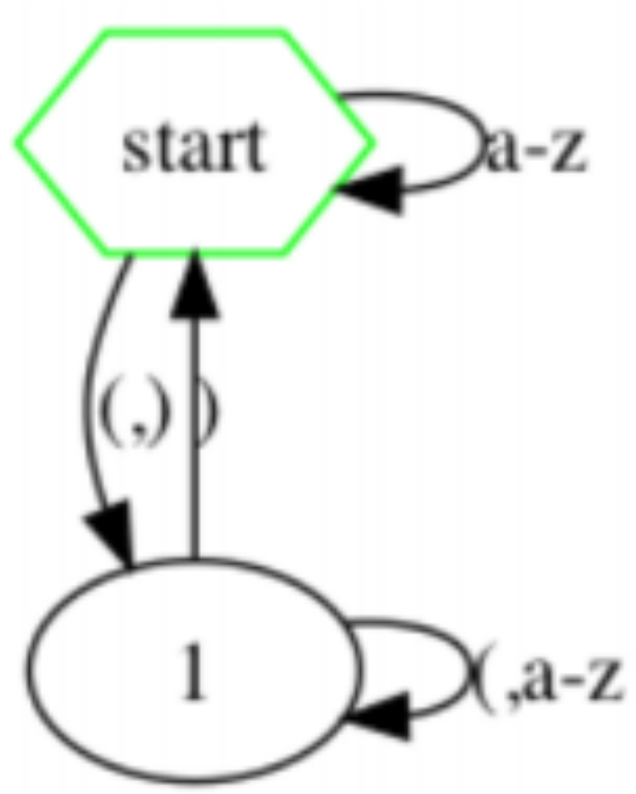
Balanced Parenthesis

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

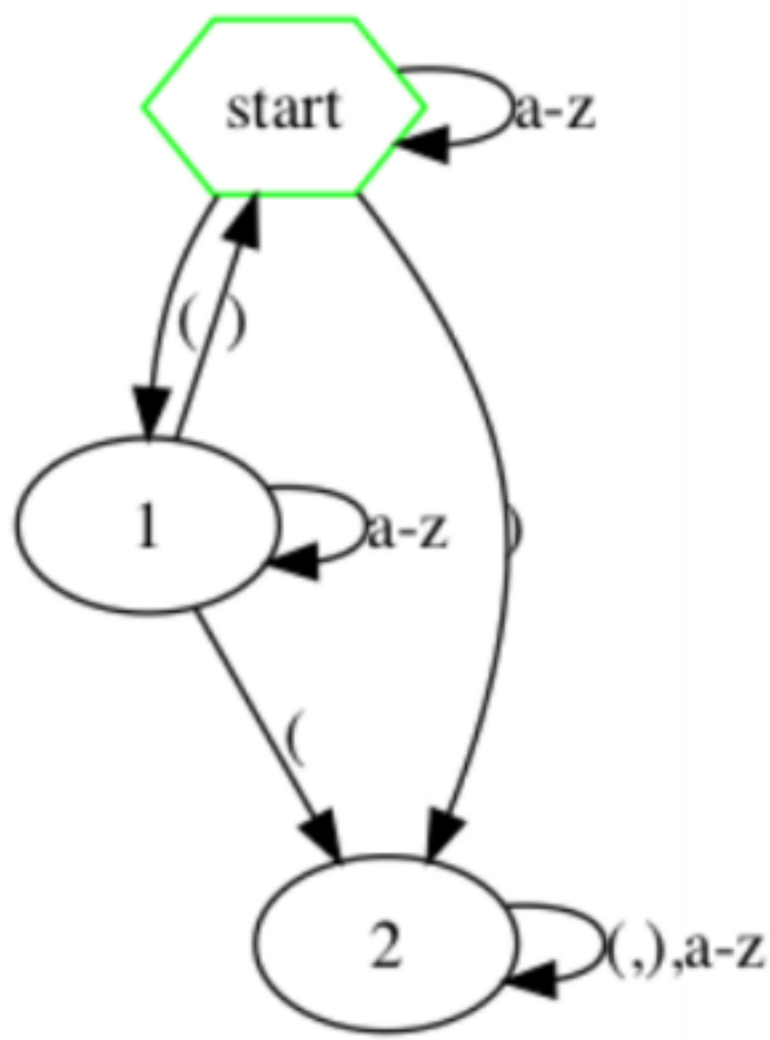
alphabet: `a-z ( )`

nesting level up to 8.

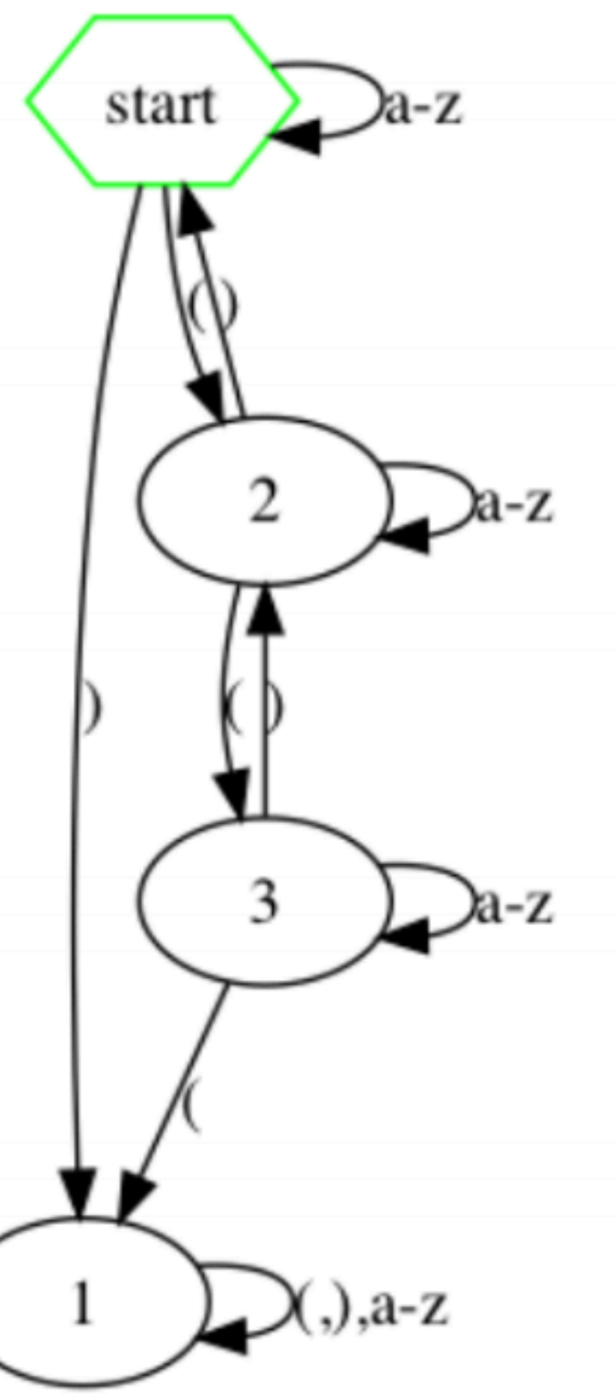
Balanced Parenthesis



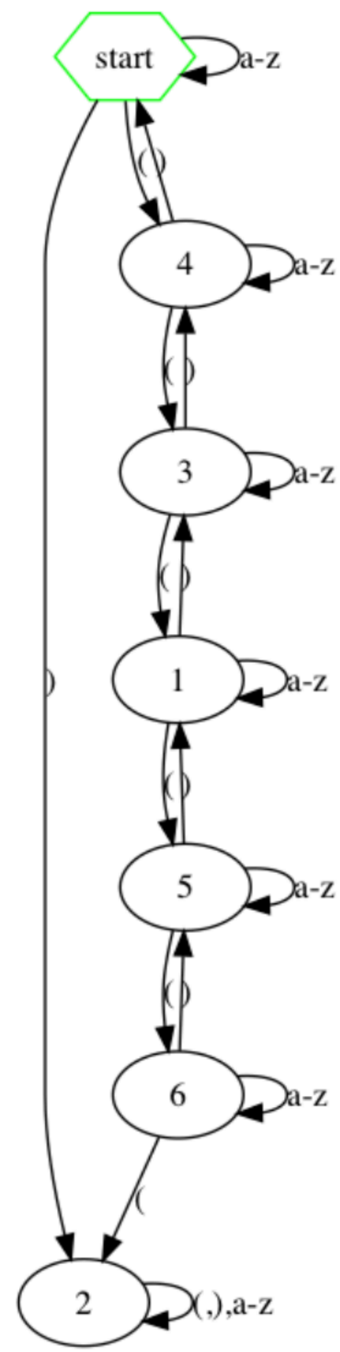
Balanced Parenthesis



Balanced Parenthesis

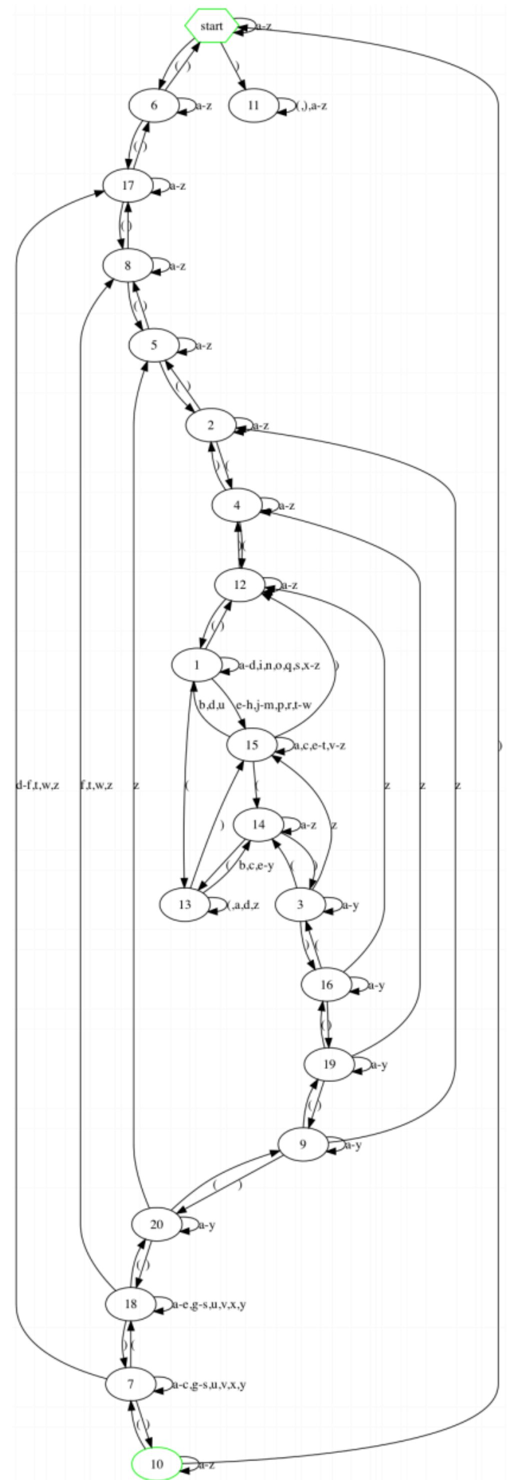


Balanced Parenthesis



Balanced Parenthesis

final automaton:

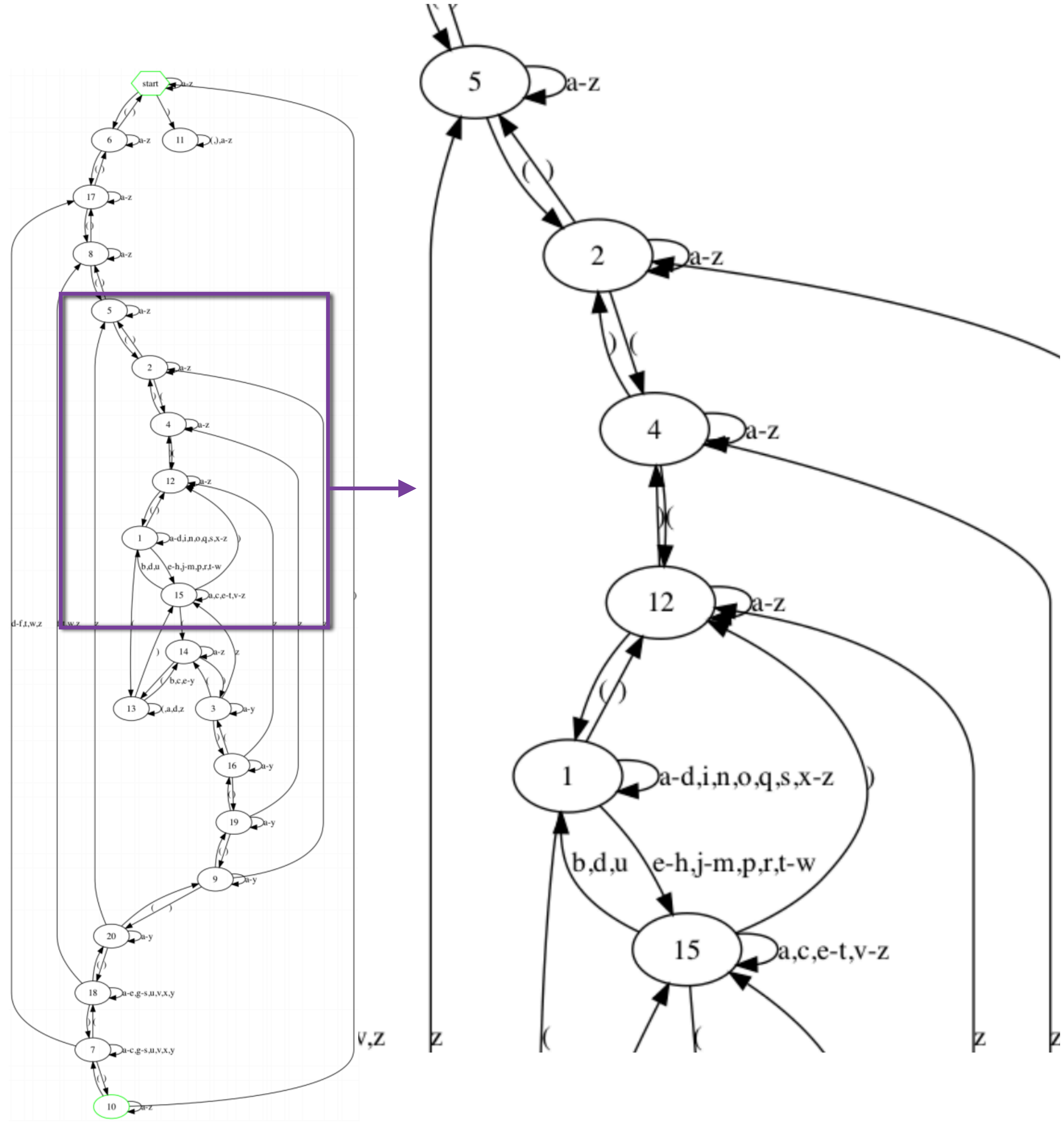


BIU NLP

TECHNION
Israel Institute of Technology

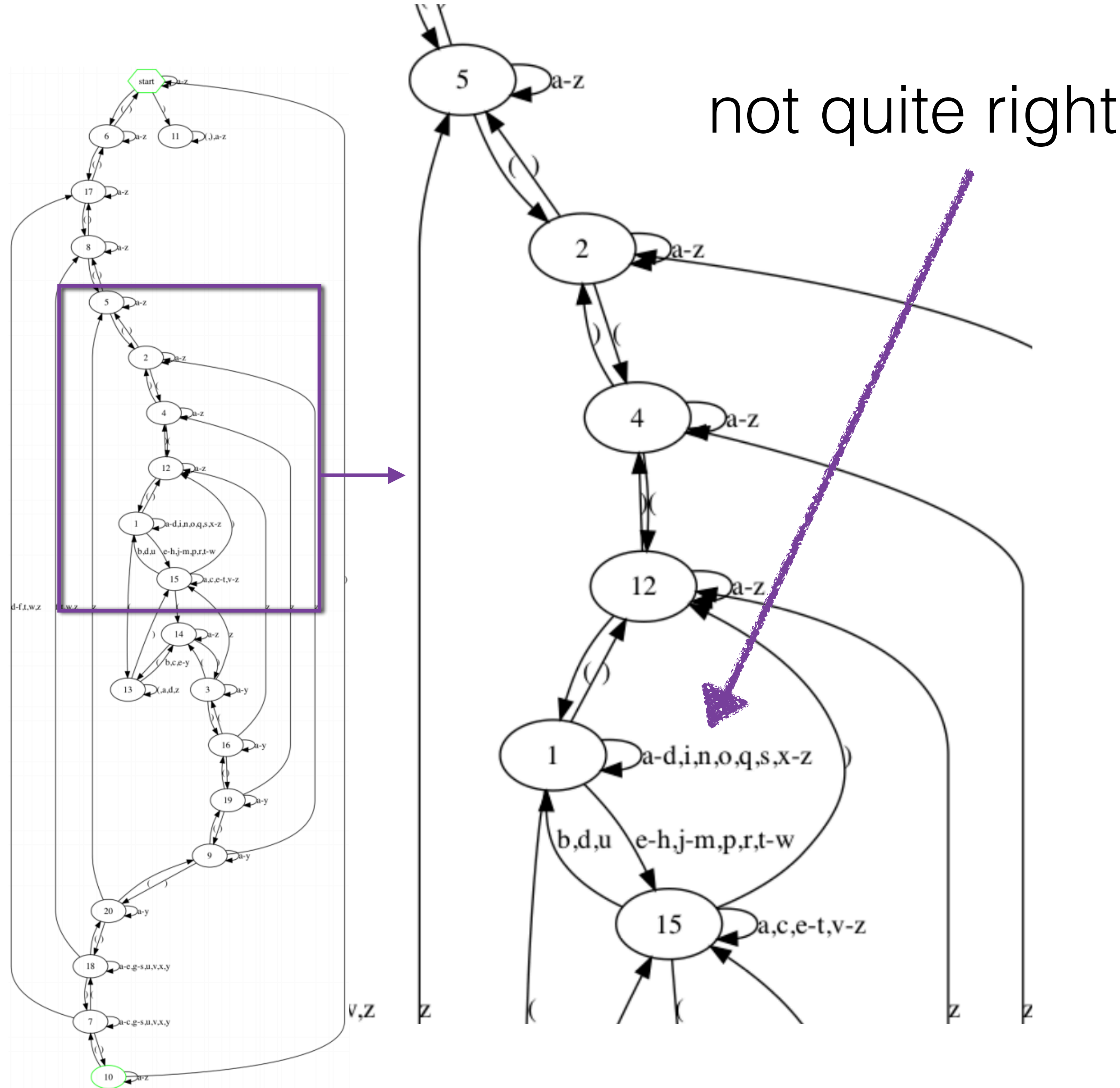
Balanced Parenthesis

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.

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 (the talk)

- LSTMs (deep nets, RNNs, ...) 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.
  - **Very excited to see the evolving community in this workshop! Keep it up!**