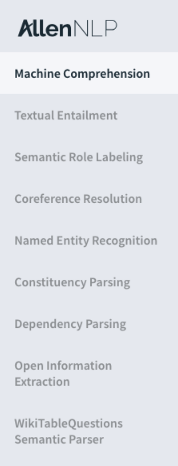
Towards Interpretable Deep Learning for Natural Language Processing

Roy Schwartz

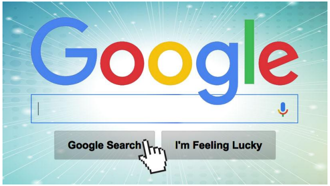
University of Washington & Allen Institute for Artificial Intelligence

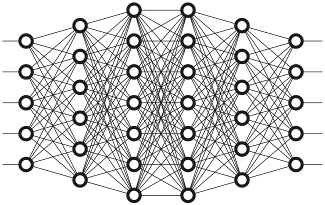
December 2018

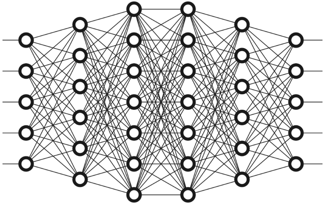
(Deep-Learning-Based) AI Today



Microsoft Translator

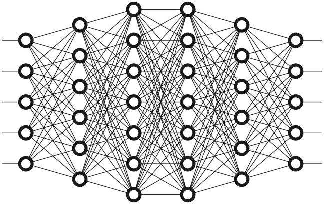






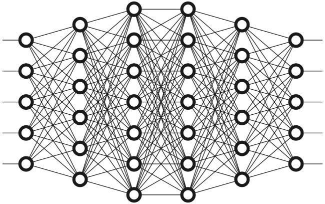
Deep Learning

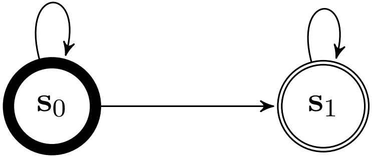
- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**

Deep Learning

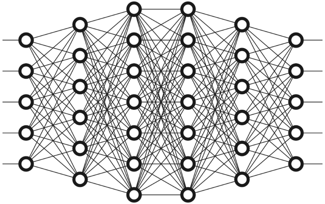
- 👍 backpropagation
- 👍 stochastic gradient descent
- 👍 PyTorch, TensorFlow, AllenNLP
- 👍 **state-of-the-art**
- 👎 architecture engineering

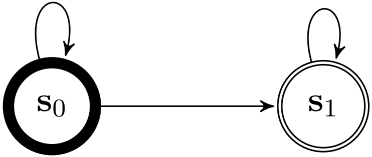




Deep Learning

- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**
- architecture engineering



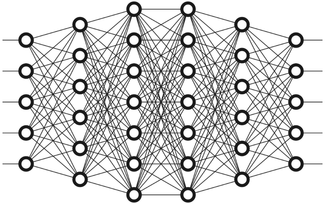


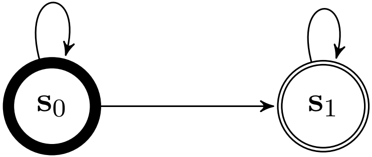
Deep Learning

- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**
- architecture engineering

Weighted Finite-State Automata

- widely studied
- understandable
- interpretable
- **informed model development**



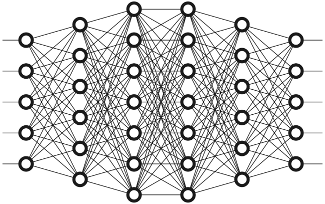


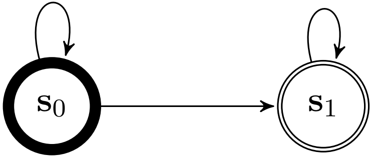
Deep Learning

- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**
- architecture engineering

Weighted Finite-State Automata

- widely studied
- understandable
- interpretable
- **informed model development**
- low performance



Deep Learning

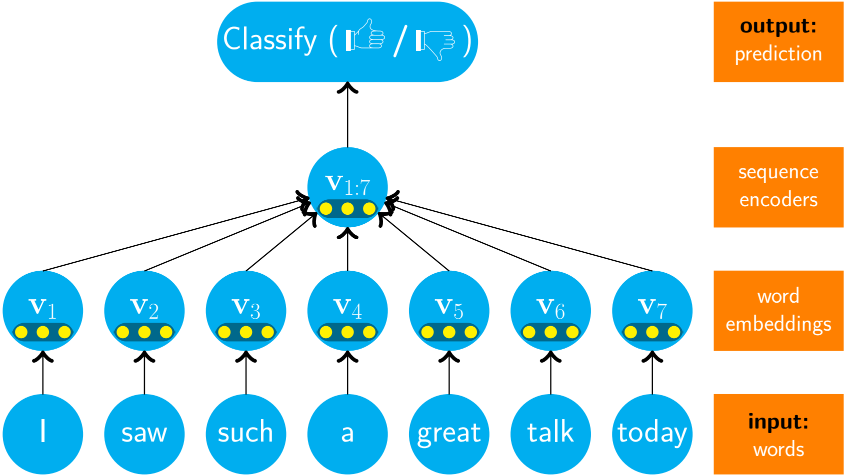
- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**
- ~~architecture engineering~~

Weighted Finite-State Automata

- widely studied
- understandable
- interpretable
- **informed model development**
- ~~low performance~~

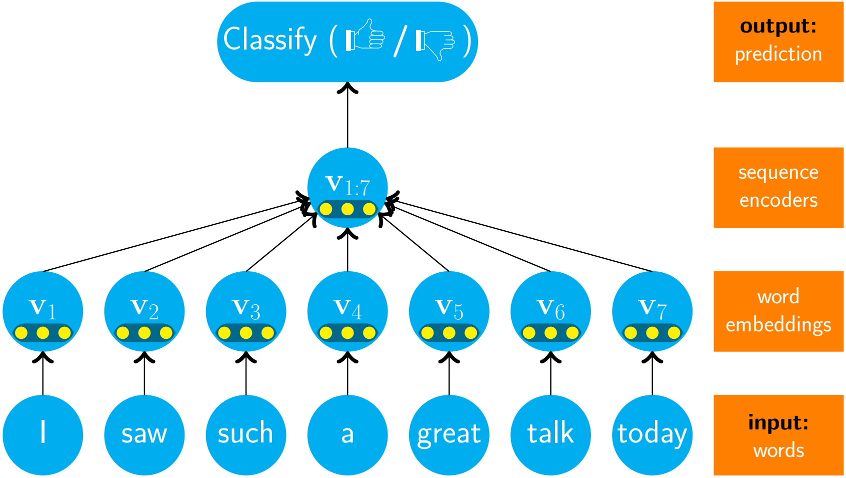
Deep Learning Models for NLP: Overview

Case Study: Sentiment Analysis



Deep Learning Models for NLP: Overview

Case Study: Sentiment Analysis

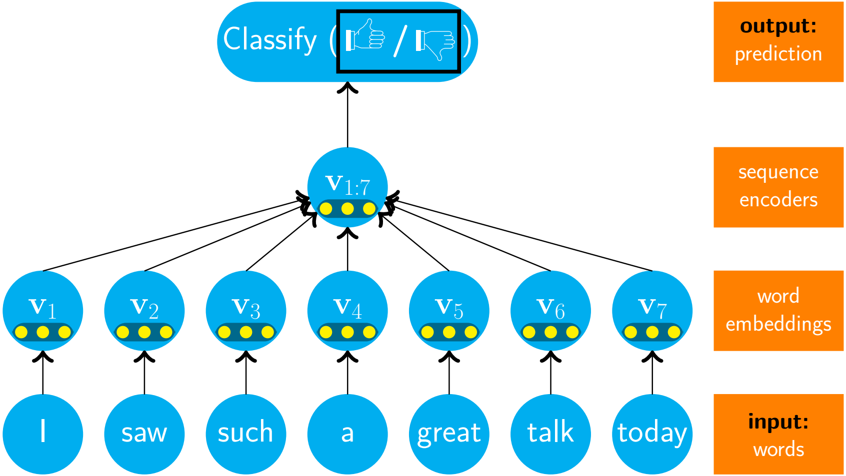


Main component in

- Machine translation
- Question answering
- Text summarization
- Sentiment analysis
- Information extraction
- ...

Deep Learning Models for NLP: Overview

Case Study: Sentiment Analysis



Main component in

- Machine translation
- Question answering
- Text summarization
- **Sentiment analysis**
- Information extraction
- ...

Overview

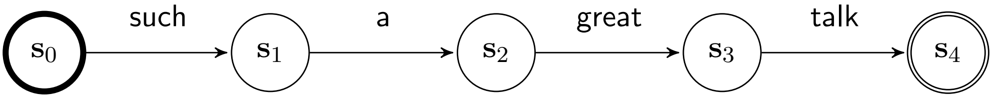
- Background: Weighted Finite-State Automata
- Neural Weighted Finite-State Automata
- Existing Deep Models as Weighted Finite-State Automata
  - Case Study: Convolutional neural networks

Overview

- **Background: Weighted Finite-State Automata**
- Neural Weighted Finite-State Automata
- Existing Deep Models as Weighted Finite-State Automata
  - Case Study: Convolutional neural networks

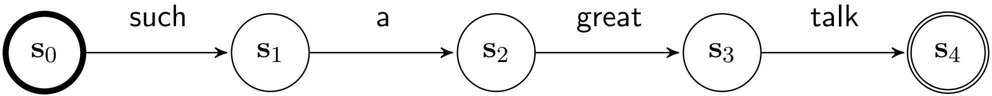
Background: Finite-State Automata

Regular Expressions (Patterns)



Background: Finite-State Automata

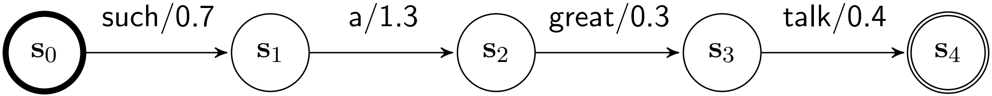
Regular Expressions (Patterns)



Pattern: *such a great talk*

Background: **Weighted** Finite-State Automata (WFSA)

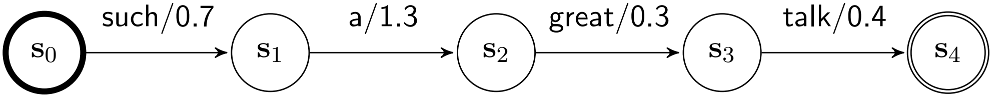
Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
    - Weights are typically pre-specified

Background: **Weighted** Finite-State Automata (WFSA)

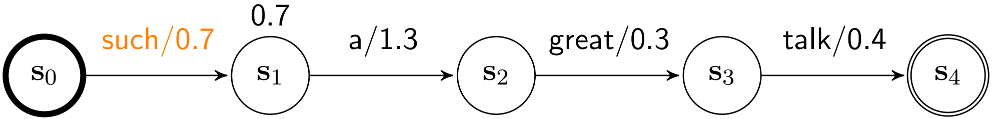
Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
  - Weights are typically pre-specified
- The **score** of a sequence is the sum of transition scores

Background: **Weighted** Finite-State Automata (WFSA)

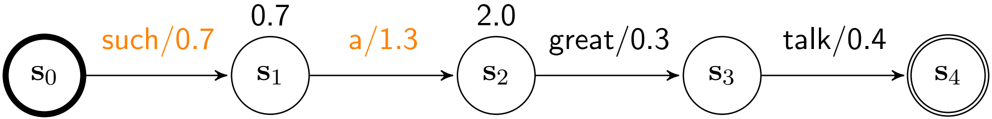
Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
  - Weights are typically pre-specified
- The **score** of a sequence is the sum of transition scores

Background: **Weighted** Finite-State Automata (WFSA)

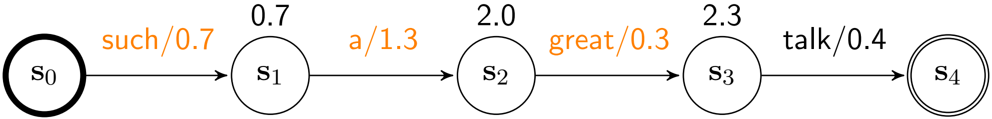
Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
  - Weights are typically pre-specified
- The **score** of a sequence is the sum of transition scores

Background: **Weighted** Finite-State Automata (WFSA)

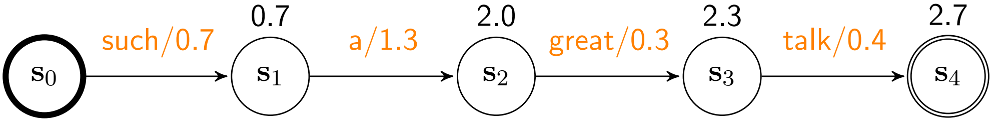
Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
  - Weights are typically pre-specified
- The **score** of a sequence is the sum of transition scores

Background: **Weighted** Finite-State Automata (WFSA)

Each Transition Defines a Weight Function



- (Weighted) pattern: *such a great talk*
  - Weights are typically pre-specified
- The **score** of a sequence is the sum of transition scores

Overview

- Background: Weighted Finite-State Automata
- Neural Weighted Finite-State Automata
- Existing Deep Models as Weighted Finite-State Automata
  - Case Study: Convolutional neural networks

Overview

- Background: Weighted Finite-State Automata
- **Neural Weighted Finite-State Automata**
- Existing Deep Models as Weighted Finite-State Automata
  - Case Study: Convolutional neural networks

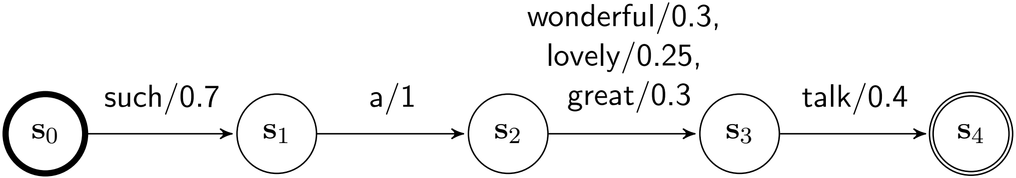
Motivation: **Soft** Pattern Matching

- *such a great talk*
  - such a *wonderful* talk, such a *lovely* talk

Motivation: **Soft** Pattern Matching

- *such a great talk*
    - such a *wonderful* talk, such a *lovely* talk

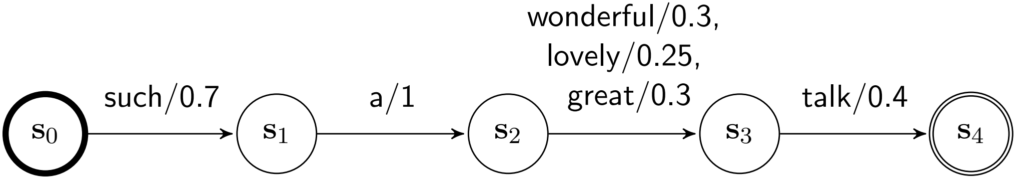
- Naive solution:



Motivation: **Soft** Pattern Matching

- *such a great talk*
  - such a *wonderful* talk, such a *lovely* talk

- Naive solution:



- Problem: **not scalable**
  - *what* a great talk, such *an awesome* talk

Solution: **Neural** Transitions

Schwartz et al., ACL 2018

Solution: **Neural** Transitions

Schwartz et al., ACL 2018

- Step 1: word $\rightarrow \mathbb{R}^d$
  - Word embeddings
  - Similar words are encoded in similar vectors

Solution: **Neural** Transitions

Schwartz et al., ACL 2018

- Step 1: word $\rightarrow \mathbb{R}^d$
  - Word embeddings
  - Similar words are encoded in similar vectors
- Step 2: Accept **all** word vectors

Solution: **Neural** Transitions

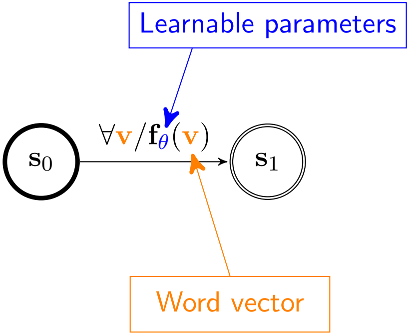
Schwartz et al., ACL 2018

- Step 1: word $\to \mathbb{R}^d$
  - Word embeddings
  - Similar words are encoded in similar vectors
- Step 2: Accept **all** word vectors
- Step 3: weights: $\mathbf{f}_\theta : \mathbb{R}^d \to \mathbb{R}$
  - These functions **favor** specific words
  - θ parameters are learned

Solution: **Neural** Transitions

Schwartz et al., ACL 2018

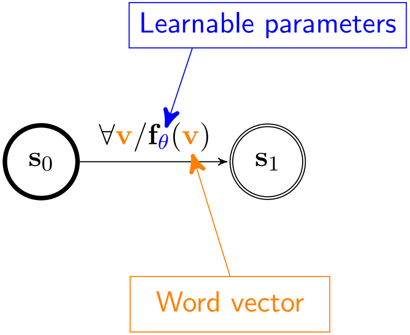
- Neural transitions **accept** all words,
- but **favor** specific words



Solution: **Neural** Transitions

Schwartz et al., ACL 2018

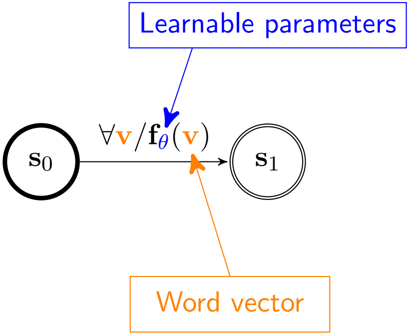
- Neural transitions **accept** all words,
- but **favor** specific words
- Example 1: *great*
  - high score: **great**, awesome, good
  - low score: bad, child, three



Solution: **Neural** Transitions

Schwartz et al., ACL 2018

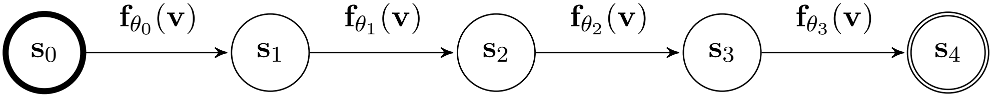
- Neural transitions **accept** all words,
- but **favor** specific words
- Example 1: *great*
  - high score: **great**, awesome, good
  - low score: bad, child, three
- Example 2: the
  - high score: **the**, a, an
  - low score: car, love, well



Neural Weighted Finite-State Automata

Schwartz et al., ACL 2018

v – word vectors
$\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$ – learned parameters



- Neural WFSAs accept **any sequence**,[1] but **prefer** certain sequences

[1] Pending length constraints

Neural Weighted Finite-State Automata

Schwartz et al., ACL 2018

v – word vectors
$\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$ – learned parameters

$$\mathbf{s}_0 \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s}_1 \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s}_2 \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s}_3 \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s}_4$$

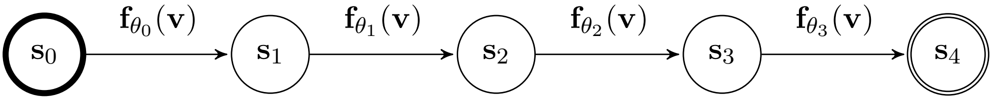
- Neural WFSAs accept **any sequence**,[1] but **prefer** certain sequences
- Example 1: *such a great talk*
  - high score: **what** a great talk, such **an awesome** talk
  - low score: such a **horrible** talk, such a **black cat**, **john went to school**

[1] Pending length constraints

Neural Weighted Finite-State Automata

Schwartz et al., ACL 2018

v – word vectors
$\theta = (\theta_0, \theta_1, \theta_2, \theta_3)$ – learned parameters



- Neural WFSAs accept **any sequence**,[1] but **prefer** certain sequences
- Example 1: *such a great talk*
  - high score: **what** a great talk, such **an awesome** talk
  - low score: such a **horrible** talk, such a **black cat**, **john went to school**
- Example 2: *is not very exciting*
  - high score: is not **particularly** exciting, **are** not very **inspiring**

[1] Pending length constraints

Training Procedure

Formally

End-to-end training:

- Input
  - $\mathbf{s}_0 \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s}_1 \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s}_2 \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s}_3 \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s}_4$
  - Word embeddings: word $\rightarrow \mathbb{R}^d$
  - Training data: pairs of $<$document, sentiment label$>$
- Output
  - Parameter values: θ

Training Procedure

Formally

End-to-end training:

- Input
  - $\mathbf{s_0} \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s_1} \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s_2} \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s_3} \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s_4}$
  - Word embeddings: word $\rightarrow \mathbb{R}^d$
  - Training data: pairs of <document, sentiment label>
- Output
  - Parameter values: θ

Test:

- Input
  - $\mathbf{s_0} \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s_1} \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s_2} \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s_3} \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s_4}$
  - Word embeddings: word $\rightarrow \mathbb{R}^d$
  - Learned parameters: θ
  - New data: <document>
- Output
  - Prediction: <sentiment label>

Training Procedure

Formally

End-to-end training:

- Input
  - $\mathbf{s}_0 \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s}_1 \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s}_2 \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s}_3 \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s}_4$
  - Word embeddings: word $\rightarrow \mathbb{R}^d$
  - Training data: pairs of <document, sentiment label>
- Output
  - Parameter values: θ

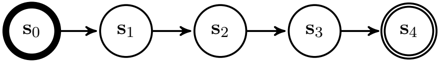
Test:

- Input
  - $\mathbf{s}_0 \xrightarrow{\mathbf{f}_{\theta_0}(\mathbf{v})} \mathbf{s}_1 \xrightarrow{\mathbf{f}_{\theta_1}(\mathbf{v})} \mathbf{s}_2 \xrightarrow{\mathbf{f}_{\theta_2}(\mathbf{v})} \mathbf{s}_3 \xrightarrow{\mathbf{f}_{\theta_3}(\mathbf{v})} \mathbf{s}_4$
  - Word embeddings: word $\rightarrow \mathbb{R}^d$
  - Learned parameters: θ
  - New data: <document>
- Output
  - Prediction: <sentiment label>

- Standard training procedure
  - Backpropagation
  - Stochastic gradient descent

Benefits of Neural WFSAs 1:

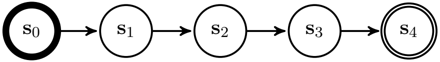
Informed Model Development



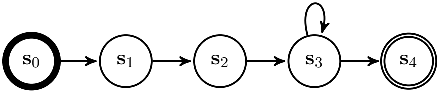
Fixed length:
such a great talk

Benefits of Neural WFSAs 1:

Informed Model Development



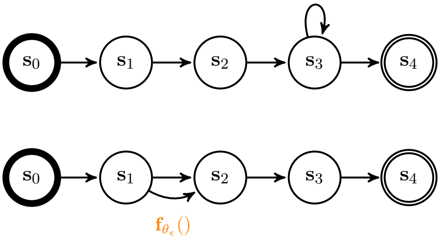
Fixed length:
such a great talk



Self loops:
such a great, wonderful, funny talk

Benefits of Neural WFSAs 1:

Informed Model Development



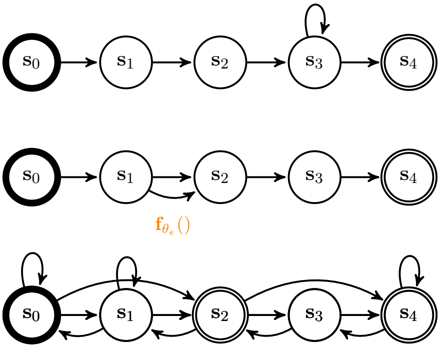
Fixed length:
such a great talk

Self loops:
such a great, wonderful, funny talk

Epsilon transitions:
such ___ *great shoes*

Benefits of Neural WFSAs 1:

Informed Model Development



Fixed length:
such a great talk

Self loops:
such a great, wonderful, funny talk

Epsilon transitions:
such __ great shoes

...

Benefits of Neural WFSAs 2:

- They are **neural**
  - Backpropagation
  - Stochastic gradient descent
  - PyTorch, TensorFlow, AllenNLP

Benefits of Neural WFSAs 2:

- They are **neural**
  - Backpropagation
  - Stochastic gradient descent
  - PyTorch, TensorFlow, AllenNLP
- Coming up:
  - Many deep models are **mathematically equivalent** to neural WFSAs
    - A (new) joint framework
    - Allows extension of these models

Overview

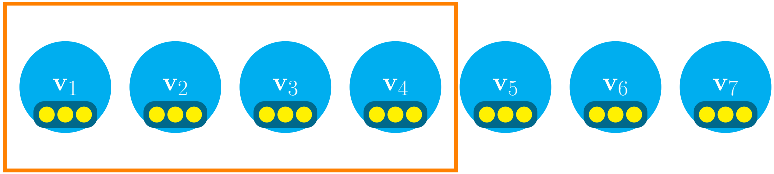
- Background: Weighted Finite-State Automata
- Neural Weighted Finite-State Automata
- Existing Deep Models as Weighted Finite-State Automata
  - Case Study: Convolutional neural networks

Overview

- Background: Weighted Finite-State Automata
- Neural Weighted Finite-State Automata
- **Existing Deep Models as Weighted Finite-State Automata**
    - **Case Study: Convolutional neural networks**

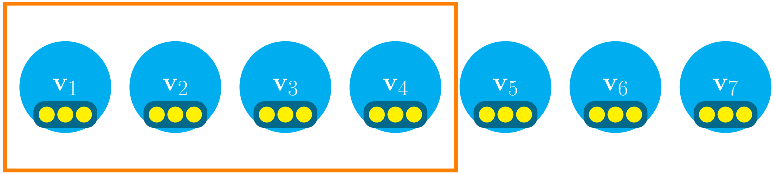
Case Study: Convolutional Neural Networks (ConvNets)

A Linear-Kernel Filter with Max-Pooling



Case Study: Convolutional Neural Networks (ConvNets)

A Linear-Kernel Filter with Max-Pooling



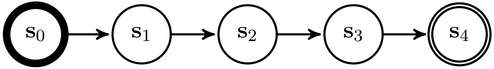
$$\boxed{S_\theta(\mathbf{v}_1 : \mathbf{v}_4) = \sum_{j=1:4} \theta_j \cdot \mathbf{v}_j}$$

Learnable parameters

Word vectors

Proposition 1: ConvNet Filters are Computing WFSA scores

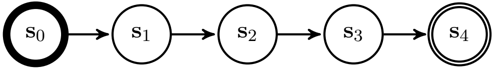
Schwartz et al., ACL 2018



Proposition 1: ConvNet Filters are Computing WFSA scores

Schwartz et al., ACL 2018

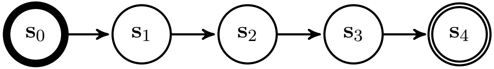
- $\mathbf{f}_{\theta_j}(\mathbf{v}) = \theta_j \cdot \mathbf{v}$



Proposition 1: ConvNet Filters are Computing WFSA scores

Schwartz et al., ACL 2018

- $\mathbf{f}_{\theta_j}(\mathbf{v}) = \theta_j \cdot \mathbf{v}$
- $s_\theta(\mathbf{v}_1 : \mathbf{v}_4) = \sum_{j=1:4} \mathbf{f}_{\theta_j}(\mathbf{v}_j) = \sum_{j=1:4} (\theta_j \cdot \mathbf{v}_j)$



ConvNets are (**Implicitly**) Computing WFSA Scores!

$$\text{ConvNet} : S_\theta(\mathbf{v}_1 : \mathbf{v}_d) = \sum_{j=1:d} (\theta_j \cdot \mathbf{v}_j) \tag{1}$$

$$\text{Neural WFSA} : s_\theta(\mathbf{v}_1 : \mathbf{v}_d) = \sum_{j=1:d} (\theta_j \cdot \mathbf{v}_j) \tag{2}$$

ConvNets are (**Implicitly**) Computing WFSA Scores!

$$\text{ConvNet}: S_\theta(\mathbf{v}_1 : \mathbf{v}_d) = \sum_{j=1:d} (\theta_j \cdot \mathbf{v}_j) \tag{1}$$

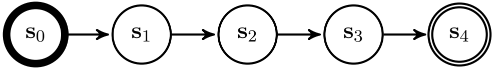
$$\text{Neural WFSA}: s_\theta(\mathbf{v}_1 : \mathbf{v}_d) = \sum_{j=1:d} (\theta_j \cdot \mathbf{v}_j) \tag{2}$$

Benefits:

- ✓ **Interpret** ConvNets
- ✓ **Improve** ConvNets

A ConvNet Learns a Fixed-Length *Soft*-Pattern!

Schwartz et al., ACL 2018



- E.g., *"such a great talk"*
  - <u>*what*</u> a great <u>*song*</u>
  - such <u>*an*</u> <u>*awesome*</u> <u>*movie*</u>

Improving ConvNets: **SoPa** (Soft-Patterns)

Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Improving ConvNets: **SoPa** (Soft-Patterns)

Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Convolutional Neural Network:

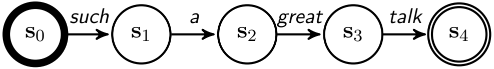
$$S_\theta(\mathbf{v}_1 : \mathbf{v}_d) = \sum_{j=1:d} (\theta_j \cdot \mathbf{v}_j)$$

Improving ConvNets: **SoPa** (Soft-Patterns)

Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Weighted Finite-State Automaton:

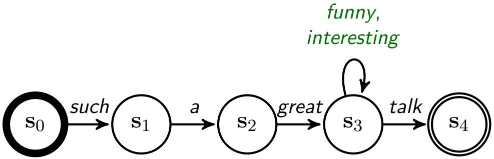


Improving ConvNets: **SoPa** (Soft-Patterns)

Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Weighted Finite-State Automaton:

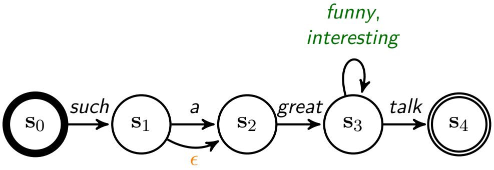


Improving ConvNets: **SoPa** (Soft-Patterns)

Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Weighted Finite-State Automaton:

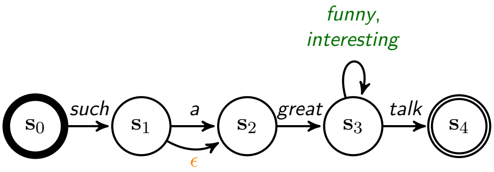


Improving ConvNets: **SoPa** (Soft-Patterns)

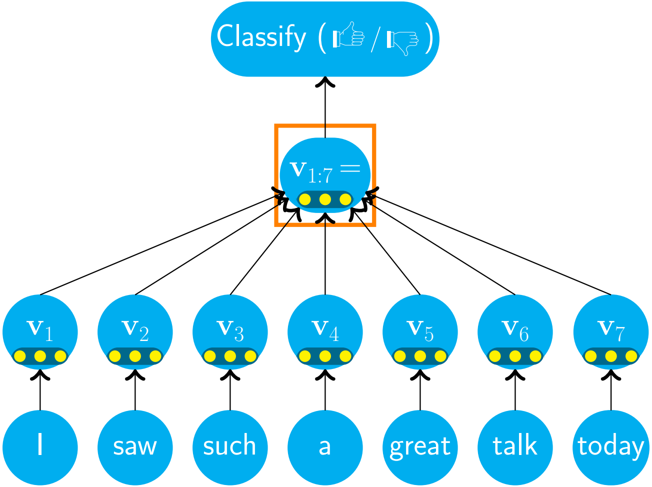
Schwartz et al., ACL 2018

- Language pattern are often flexible-length
- *such a great talk*
  - such a great, *funny, interesting* talk
  - such __ great *shoes*

Weighted Finite-State Automaton:



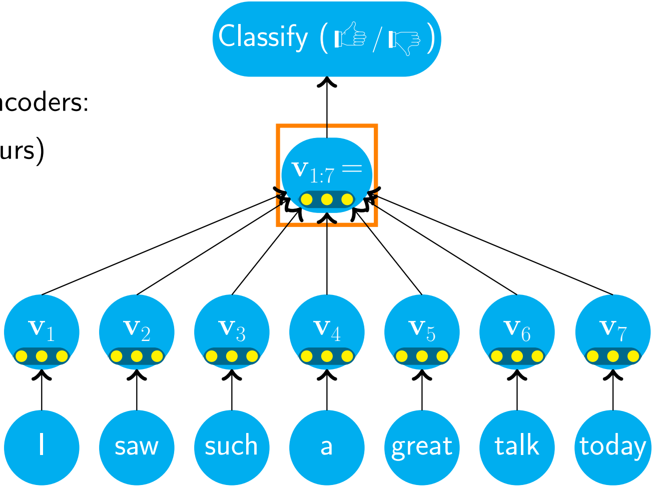
Sentiment Analysis Experiments



Sentiment Analysis Experiments

Sequence encoders:

- **SoPa** (ours)
- ConvNet

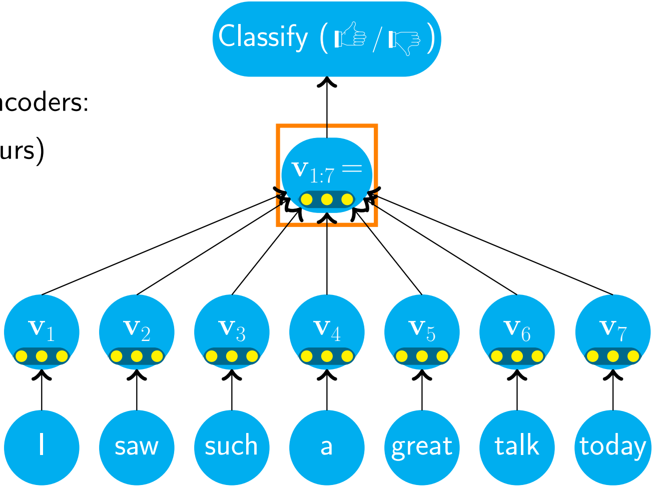


Sentiment Analysis Experiments

Sequence encoders:

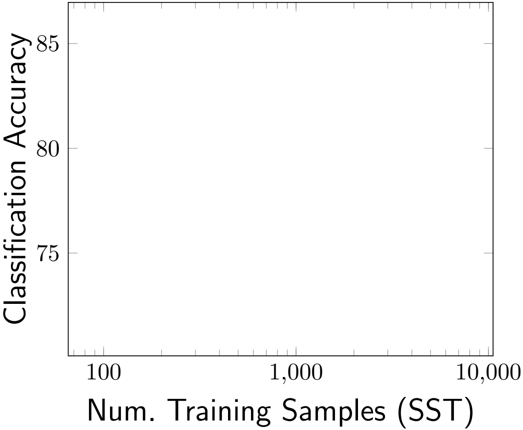
- **SoPa** (ours)
- ConvNet

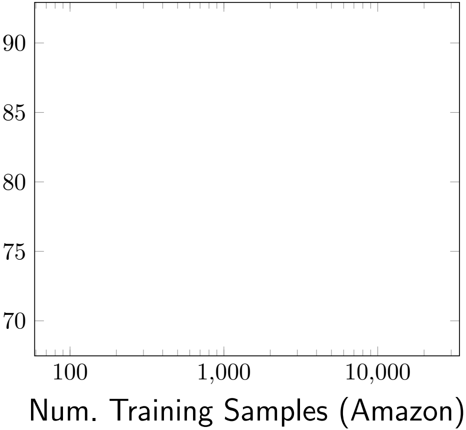
LSTM



Sentiment Analysis Results

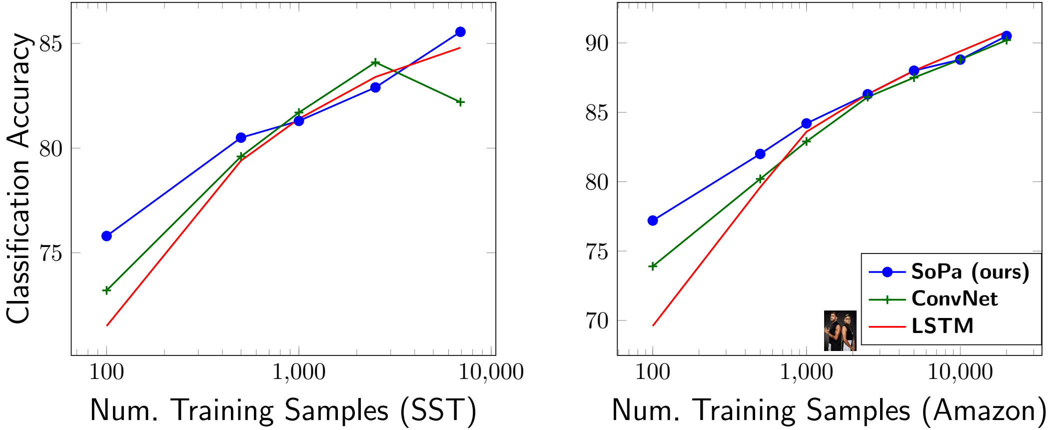
Schwartz et al., ACL 2018





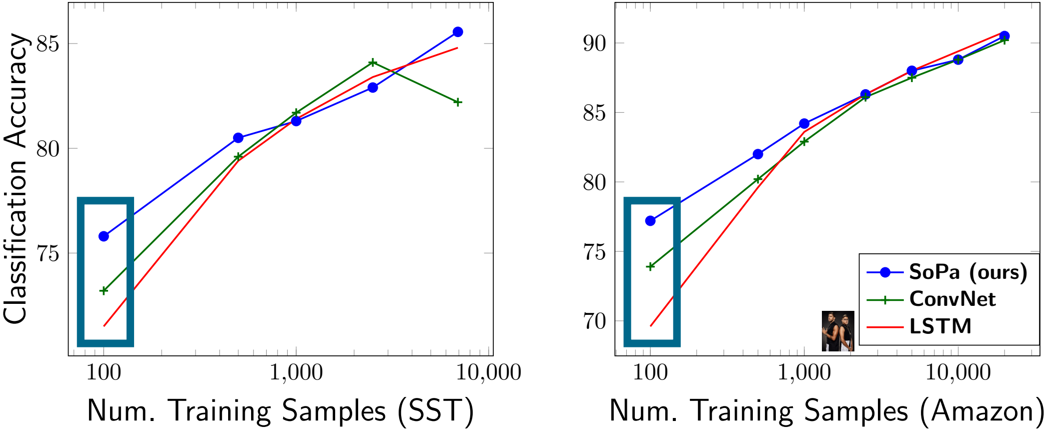
Sentiment Analysis Results

Schwartz et al., ACL 2018



Sentiment Analysis Results

Schwartz et al., ACL 2018



Interpreting SoPa

Soft Patterns!

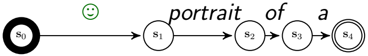
- For each learned pattern, extract the 4 top scoring phrases in the training set

Interpreting SoPa

Soft Patterns!

- For each learned pattern, extract the 4 top scoring phrases in the training set

	Highest Scoring Phrases	
Patt. 1	mesmerizing	portrait of a
	engrossing	portrait of a
	clear-eyed	portrait of an
	fascinating	portrait of a



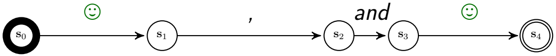
Interpreting SoPa

Soft Patterns!

- For each learned pattern, extract the 4 top scoring phrases in the training set

	Highest Scoring Phrases	
Patt. 1	mesmerizing	portrait of a
	engrossing	portrait of a
	clear-eyed	portrait of an
	fascinating	portrait of a

	Highest Scoring Phrases		
Patt. 2	honest	,	and enjoyable
	forceful	,	and beautifully
	energetic	,	and surprisingly



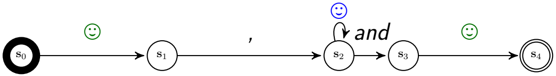
Interpreting SoPa

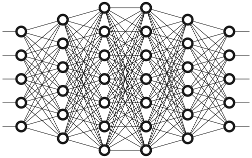
Soft Patterns!

- For each learned pattern, extract the 4 top scoring phrases in the training set

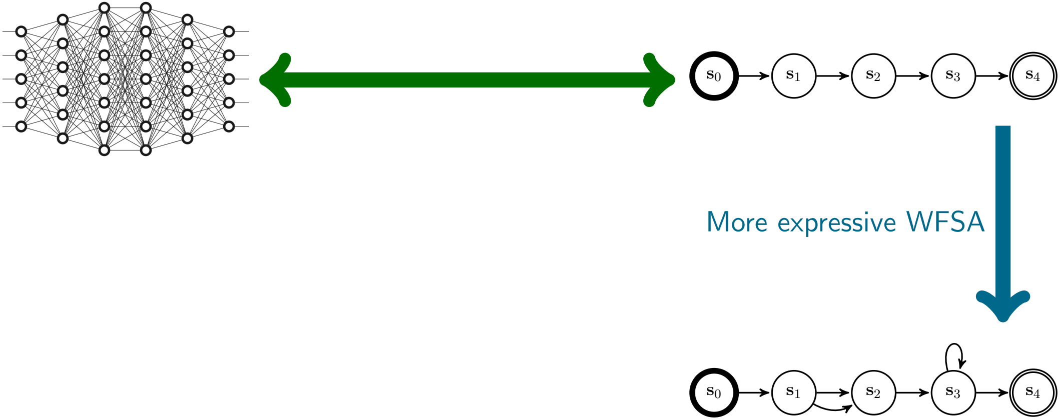
	Highest Scoring Phrases			
Patt. 1	mesmerizing	portrait	of	a
	engrossing	portrait	of	a
	clear-eyed	portrait	of	an
	fascinating	portrait	of	a

	Highest Scoring Phrases				
Patt. 2	honest	,		and	enjoyable
	forceful	,		and	beautifully
	energetic	,		and	surprisingly
	unpretentious	,	charming$_{SL}$	,	quirky

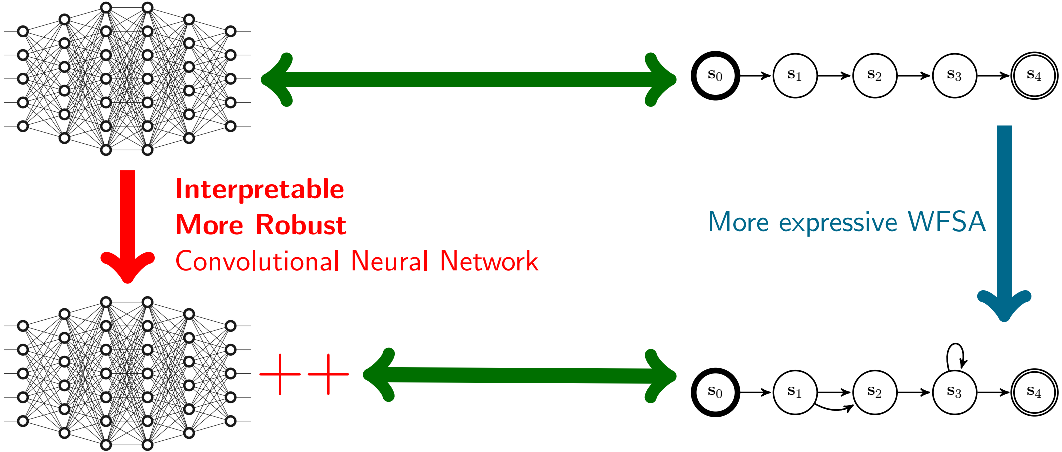





$\mathbf{s}_0$
$\mathbf{s}_1$
$\mathbf{s}_2$
$\mathbf{s}_3$
$\mathbf{s}_4$



$\mathbf{s}_0$ $\mathbf{s}_1$ $\mathbf{s}_2$ $\mathbf{s}_3$ $\mathbf{s}_4$
More expressive WFSA
$\mathbf{s}_0$ $\mathbf{s}_1$ $\mathbf{s}_2$ $\mathbf{s}_3$ $\mathbf{s}_4$


Interpretable
More Robust
Convolutional Neural Network
More expressive WFSA
++
s_0
s_1
s_2
s_3
s_4

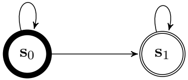
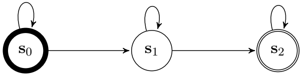
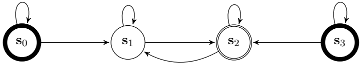
Many Existing Deep Models are Neural WFSAs!

Peng, **Schwartz** et al., EMNLP 2018

Mikolov et al. Balduzzi and Ghifary Bradbury et al. Lei et al.	arXiv 2014 ICML 2016 ICLR 2017 EMNLP 2018	
Lei et al.	NAACL 2016	
Foerster et al.	ICML 2017	

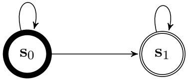
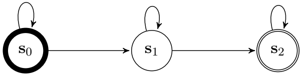
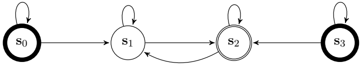
Many Existing Deep Models are Neural WFSAs!

Peng, **Schwartz** et al., EMNLP 2018

<table>
<tr><td>Mikolov et al.
Balduzzi and Ghifary
Bradbury et al.
Lei et al.</td><td>arXiv 2014
ICML 2016
ICLR 2017
EMNLP 2018</td><td></td></tr>
<tr><td>Lei et al.</td><td>NAACL 2016</td><td></td></tr>
<tr><td>Foerster et al.</td><td>ICML 2017</td><td></td></tr>
</table>

Many Existing Deep Models are Neural WFSAs!

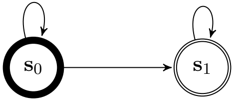
Peng, **Schwartz** et al., EMNLP 2018

Mikolov et al. Balduzzi and Ghifary Bradbury et al. Lei et al.	arXiv 2014 ICML 2016 ICLR 2017 EMNLP 2018	
Lei et al.	NAACL 2016	
Foerster et al.	ICML 2017	

- Six recent recurrent neural networks (RNN) models are also **implicitly** computing WFSA scores

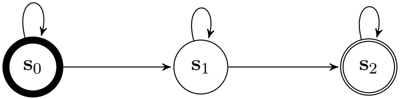
Developing more Robust WFSA Models

$\mathcal{S}_2$:



Mikolov et al. (2014)
Balduzzi and Ghifary (2016)
Bradbury et al. (2017)
Lei et al. (2018)

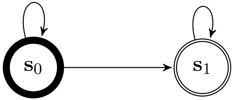
$\mathcal{S}_3$:



Lei et al. (2016)

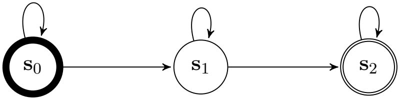
Developing more Robust WFSA Models

$\mathcal{S}_2$:



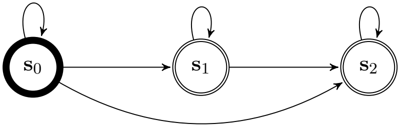
Mikolov et al. (2014)
Balduzzi and Ghifary (2016)
Bradbury et al. (2017)
Lei et al. (2018)

$\mathcal{S}_3$:



Lei et al. (2016)

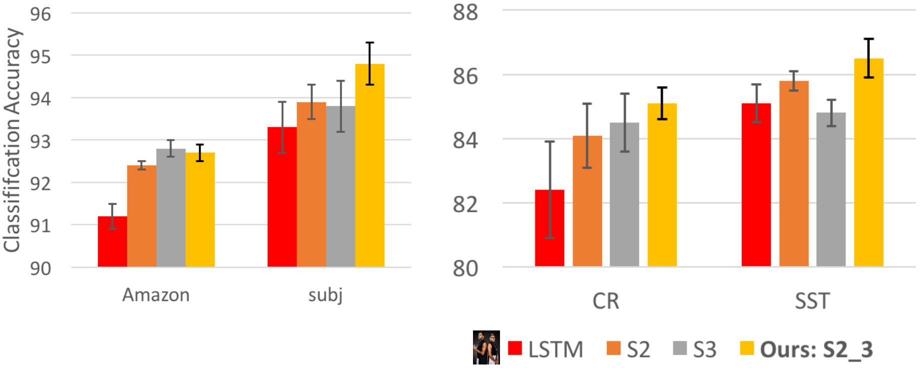
$\mathcal{S}_{2\_3}$:



Peng, **Schwartz** et al. (2018)

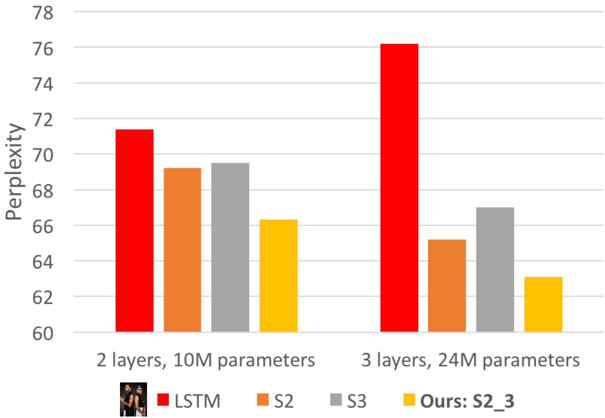
Sentiment Analysis Results

Peng, **Schwartz** et al., EMNLP 2018

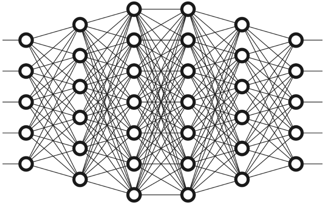
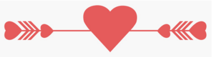
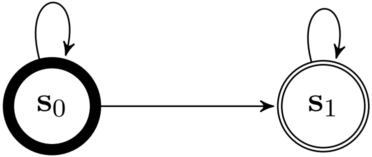


Language Modeling Results

Peng, **Schwartz** et al., EMNLP 2018



lower is better

Deep Learning

- backpropagation
- stochastic gradient descent
- PyTorch, TensorFlow, AllenNLP
- **state-of-the-art**
- ~~architecture engineering~~

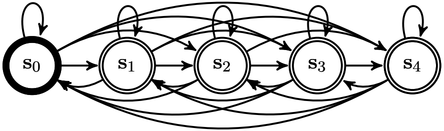
Weighted Finite-State Automata

- widely studied
- understandable
- interpretable
- **informed model development**
- ~~low performance~~

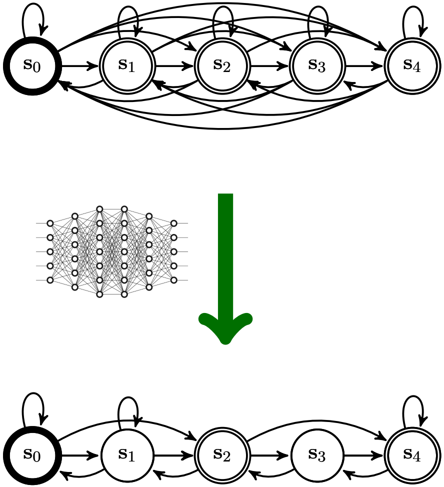
Work in Progress 1: Are All Deep Models for NLP Equivalent to WFSAs?

- Elman RNN: $\mathbf{h}_i = \sigma(\mathbf{W}\mathbf{h}_{i-1} + \mathbf{U}\mathbf{v}_i + \mathbf{b})$
- The interaction between $\mathbf{h}_i$ and $\mathbf{h}_{i-1}$ is via affine transformations followed by nonlinearities
  - Same for LSTM 
- Most probably **not** equivalent to a WFSA

Work in Progress 2: **Automatic** Model Development

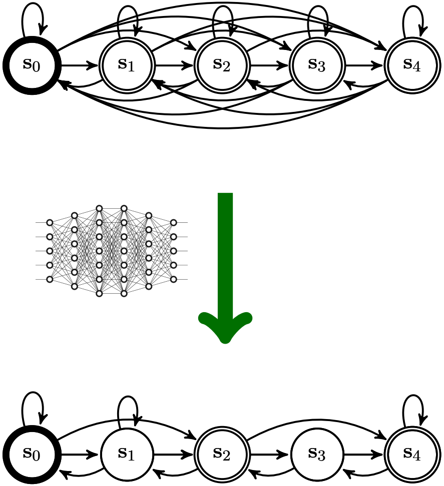


Work in Progress 2: **Automatic** Model Development



Work in Progress 2: **Automatic** Model Development

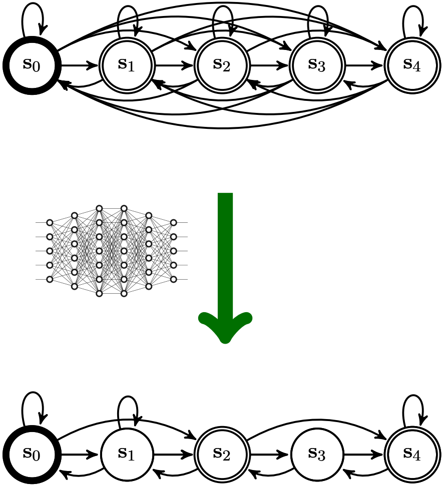
Deep learning: model engineering



Work in Progress 2: **Automatic** Model Development

Deep learning: model engineering

SoPa: *informed* model development

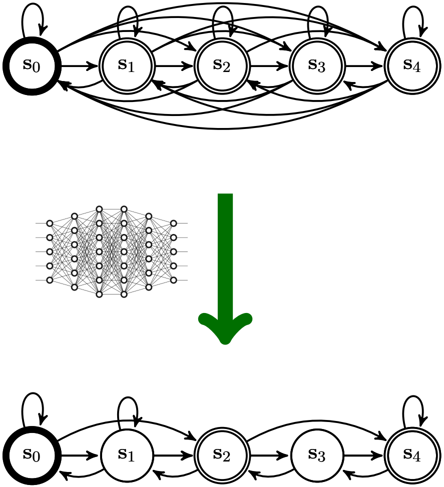


Work in Progress 2: **Automatic** Model Development

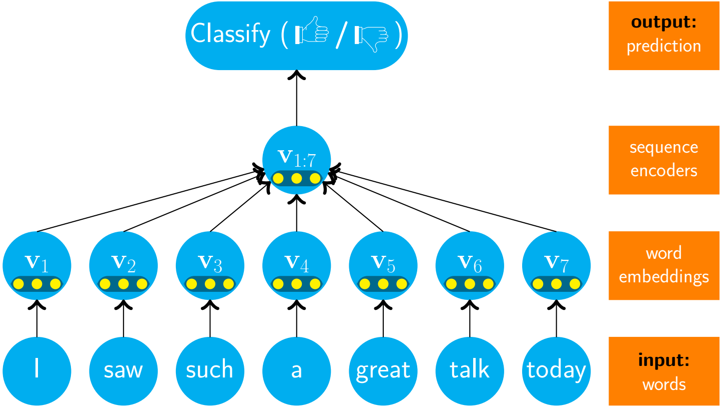
Deep learning: model engineering

SoPa: *informed* model development

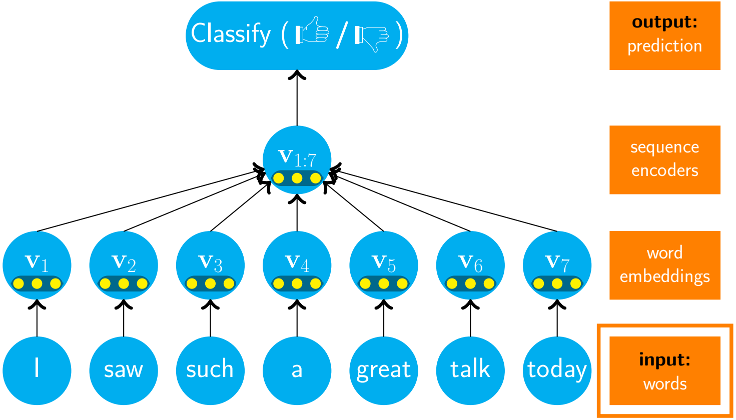
New: **Automatic** model development



Other Projects

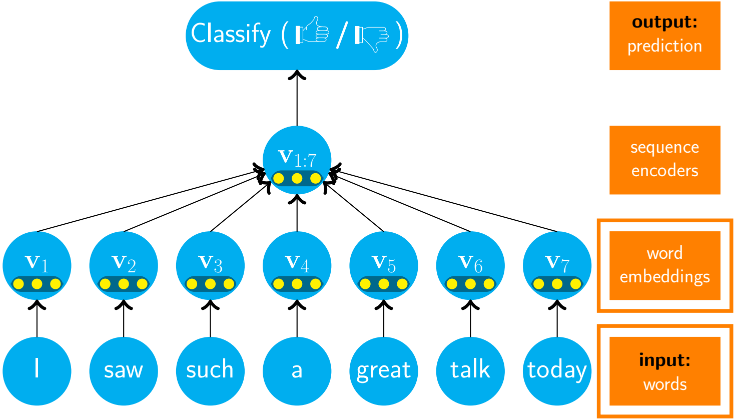


Other Projects



Schwartz et al., EMNLP 2013
Schwartz et al., COLING 2014

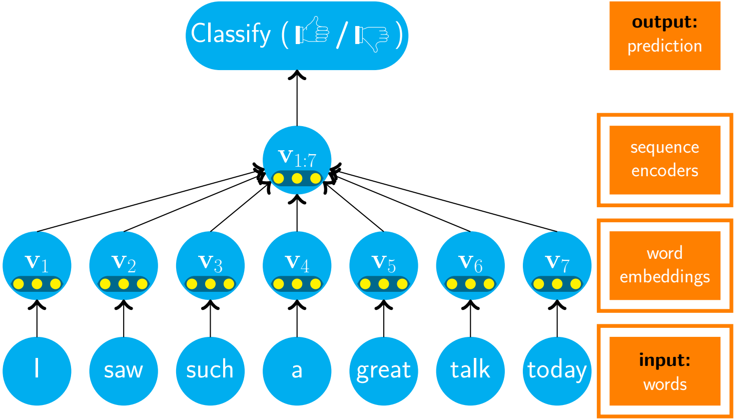
Other Projects



Schwartz et al., CoNLL 2015
Rubinstein et al., ACL 2015
Schwartz et al., NAACL 2016
Vulić et al., CoNLL 2017
Peters et al., 2018

Schwartz et al., EMNLP 2013
Schwartz et al., COLING 2014

Other Projects



Schwartz et al., ACL 2018
Peng et al., EMNLP 2018
Liu et al., RepL4NLP 2018
best paper award

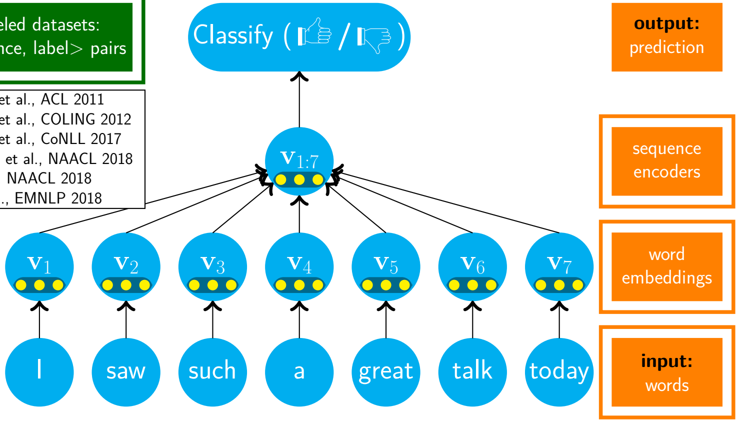
Schwartz et al., CoNLL 2015
Rubinstein et al., ACL 2015
Schwartz et al., NAACL 2016
Vulić et al., CoNLL 2017
Peters et al., 2018

Schwartz et al., EMNLP 2013
Schwartz et al., COLING 2014

Other Projects

Labeled datasets:
<sentence, label> pairs

Schwartz et al., ACL 2011
Schwartz et al., COLING 2012
Schwartz et al., CoNLL 2017
Gururangan et al., NAACL 2018
Kang et al., NAACL 2018
Zellers et al., EMNLP 2018



Schwartz et al., ACL 2018
Peng et al., EMNLP 2018
Liu et al., RepL4NLP 2018
best paper award

Schwartz et al., CoNLL 2015
Rubinstein et al., ACL 2015
Schwartz et al., NAACL 2016
Vulić et al., CoNLL 2017
Peters et al., 2018

Schwartz et al., EMNLP 2013
Schwartz et al., COLING 2014

Annotation Artifacts in NLP Datasets

Schwartz et al., CoNLL 2017; Gururangan, Swayamdipta, Levy, **Schwartz** et al., NAACL 2018

Premise	A person is running on the beach	
Hypothesis	The person is sleeping	

Textual Entailment (state-of-the-art ~90% accuracy)

Annotation Artifacts in NLP Datasets

Schwartz et al., CoNLL 2017; Gururangan, Swayamdipta, Levy, **Schwartz** et al., NAACL 2018

Premise	A person is running on the beach	
Hypothesis	The person is sleeping $\xrightarrow{?}$	**entailment** **contradiction** **neutral**

Textual Entailment (state-of-the-art ∼90% accuracy)

Annotation Artifacts in NLP Datasets

Schwartz et al., CoNLL 2017; Gururangan, Swayamdipta, Levy, **Schwartz** et al., NAACL 2018

Premise	A person is running on the beach	
Hypothesis	The person is sleeping $\xrightarrow{?}$ $\xrightarrow{?}$ $\xrightarrow{?}$	entailment **contradiction** neutral

Textual Entailment (state-of-the-art ∼90% accuracy)

AllenNLP Demo!

Annotation Artifacts in NLP Datasets

Schwartz et al., CoNLL 2017; Gururangan, Swayamdipta, Levy, **Schwartz** et al., NAACL 2018

Premise	A person is running on the beach	
Hypothesis	The person is sleeping $\xrightarrow{?}$ $\xrightarrow{?}$ $\xrightarrow{?}$	entailment **contradiction** neutral

Textual Entailment (state-of-the-art ∼90% accuracy)

- The word "sleeping" is over-represented in the training data with **contradiction** label
  - **annotation artifact**
- State-of-the-art models focus on this word rather than *understanding* the text

Annotation Artifacts in NLP Datasets

Schwartz et al., CoNLL 2017; Gururangan, Swayamdipta, Levy, **Schwartz** et al., NAACL 2018

Premise	A person is running on the beach	
Hypothesis	The person is sleeping	? entailment ? **contradiction** ? neutral

Textual Entailment (state-of-the-art ∼90% accuracy)

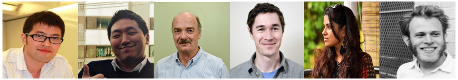
- The word "sleeping" is over-represented in the training data with **contradiction** label
  - **annotation artifact**
- State-of-the-art models focus on this word rather than *understanding* the text
- Models are **not as strong** as we think they are

Long Term Vision

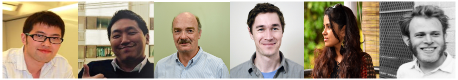
Long Term Vision

- **Explainable** models
- **Unbiased** models

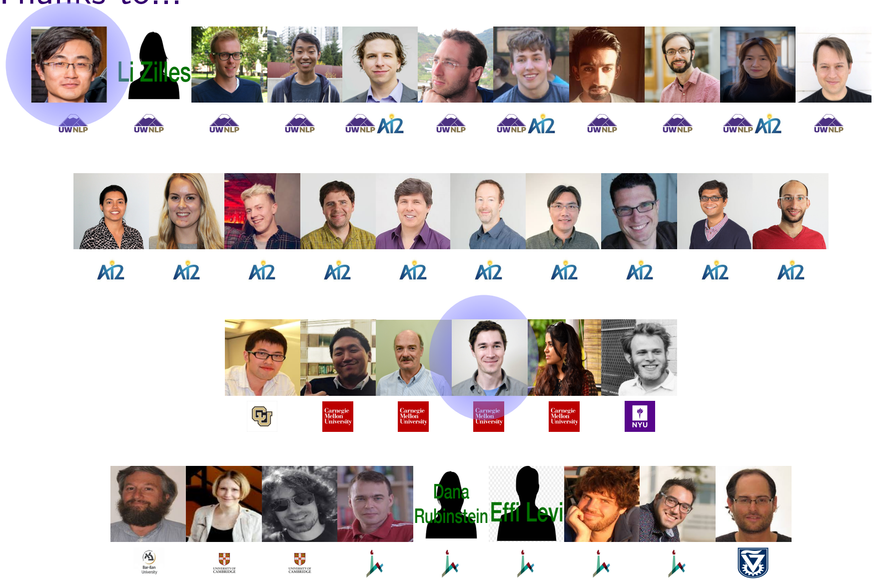
Special Thanks to...

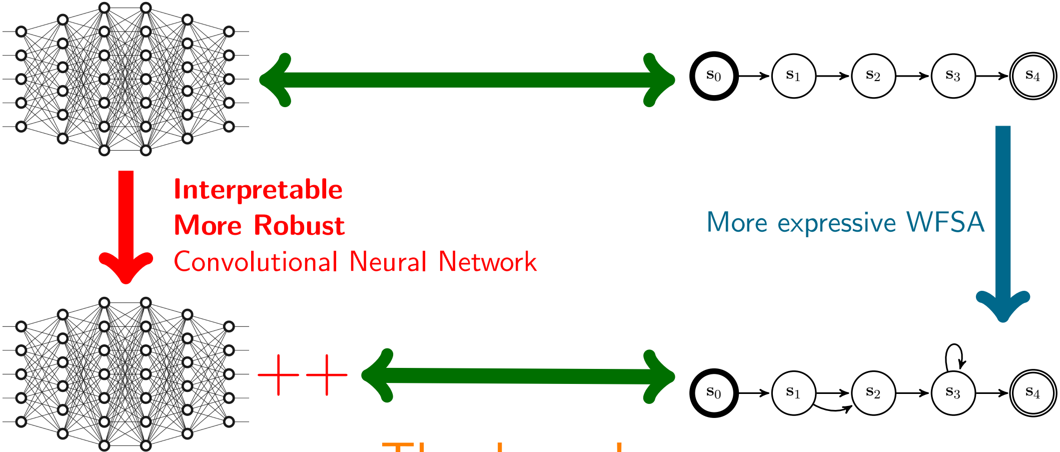


Special Thanks to...



Special Thanks to...

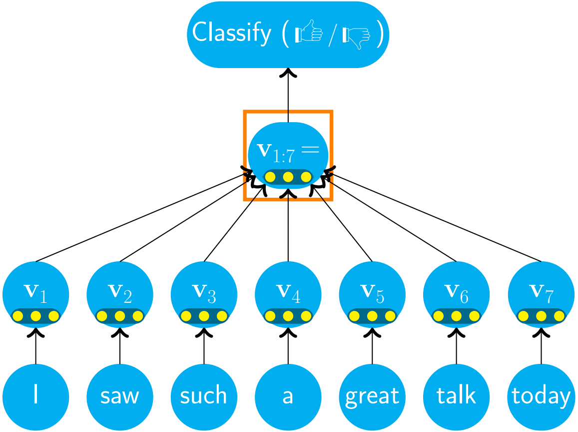




Thank you!

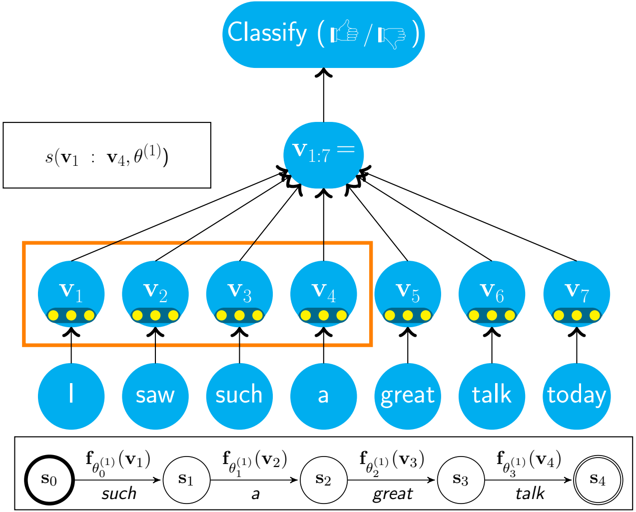
Roy Schwartz
homes.cs.washington.edu/~roysch/ roysch@cs.washington.edu

Neural WFSAs as Sequence Encoders



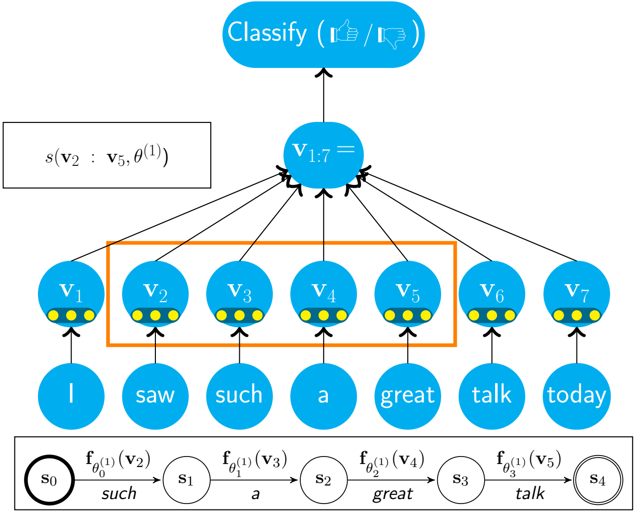
back to main

Neural WFSAs as Sequence Encoders



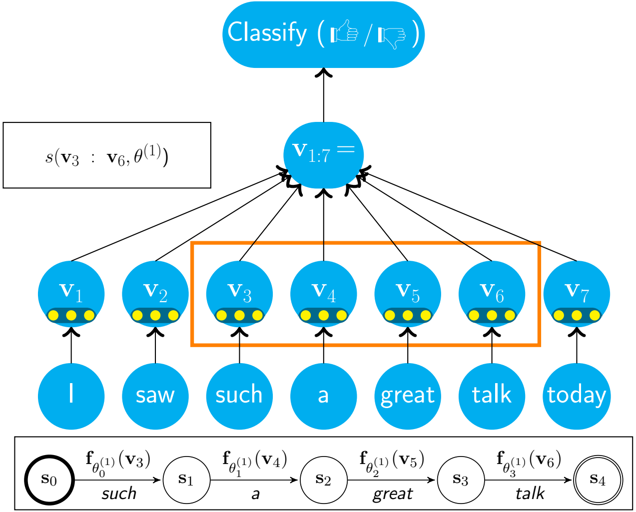
back to main

Neural WFSAs as Sequence Encoders



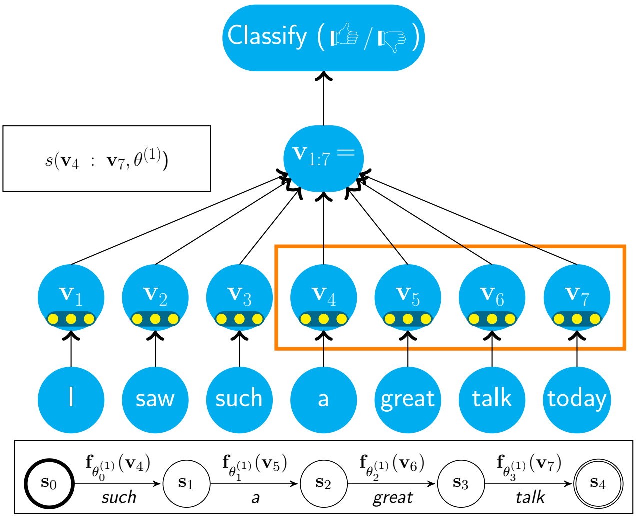
back to main

Neural WFSAs as Sequence Encoders



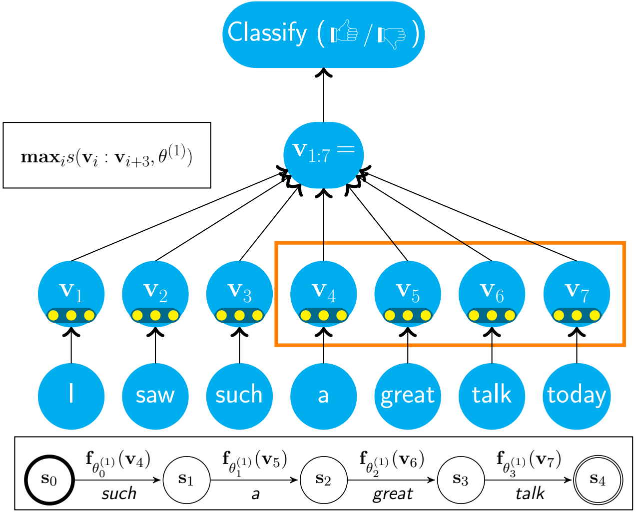
back to main

Neural WFSAs as Sequence Encoders



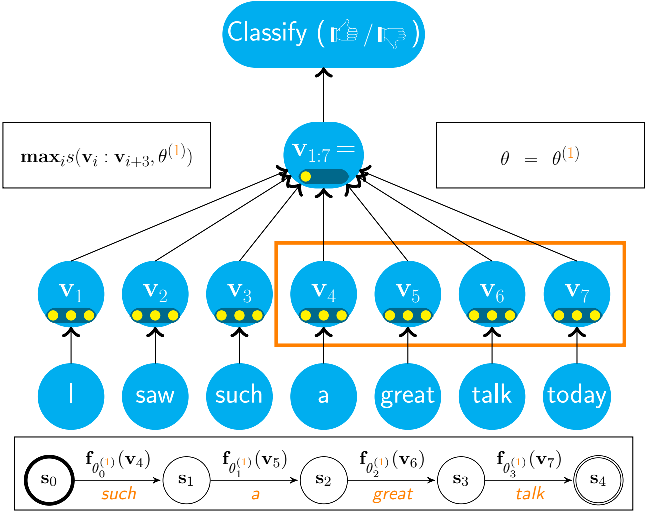
back to main

Neural WFSAs as Sequence Encoders



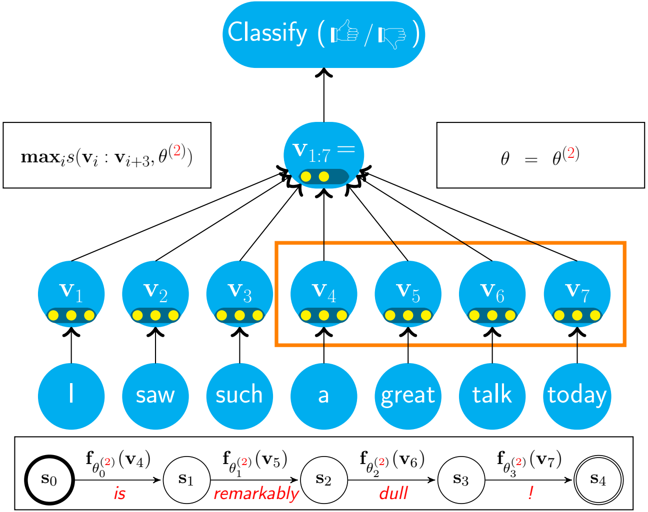
back to main

Neural WFSAs as Sequence Encoders



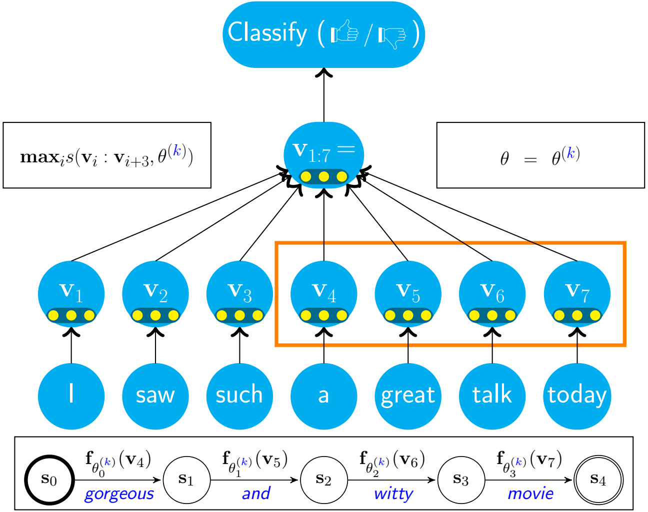
back to main

Neural WFSAs as Sequence Encoders



back to main

Neural WFSAs as Sequence Encoders



back to main

SoPa Complexity

- Running the Viterbi (1967) algorithm on a sequence of n tokens and a WFSA of d states typically takes $O(d^3 + d^2(n))$
- We only allow zero or one ϵ-transition at a time $\Rightarrow O(d^2(n))$
- We only allow self-loop and main path transitions $\Rightarrow O(dn)$
- Scores on all patterns can be computed in parallel
  - GPU optimization further reduces the observed runtime to be **sublinear** in d

back to main

Interpreting SoPa

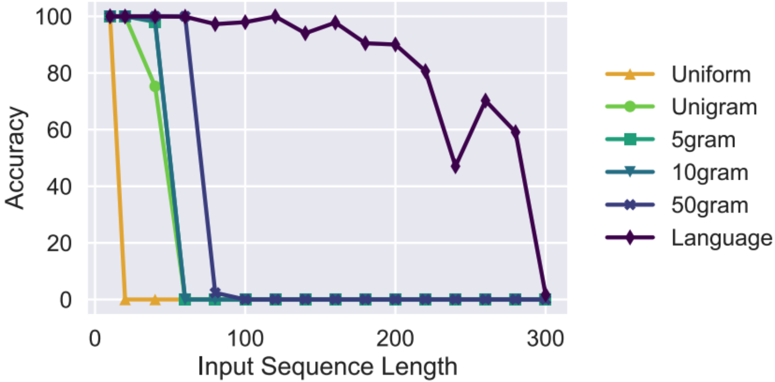
Visualizing Sentiment Predictions

- Leave-one-out method on all patterns
- Visualize the spans with the largest (**positive**) and (*negative*) contribution

Analyzed Documents
it's dumb, **but more importantly**, *it's just not scary*
While **its careful pace and** seemingly *opaque story* may not satisfy every movie-goer's appetite, the film's final scene is **soaringly, transparently moving**

LSTMs Exploit Linguistic Attributes of Data

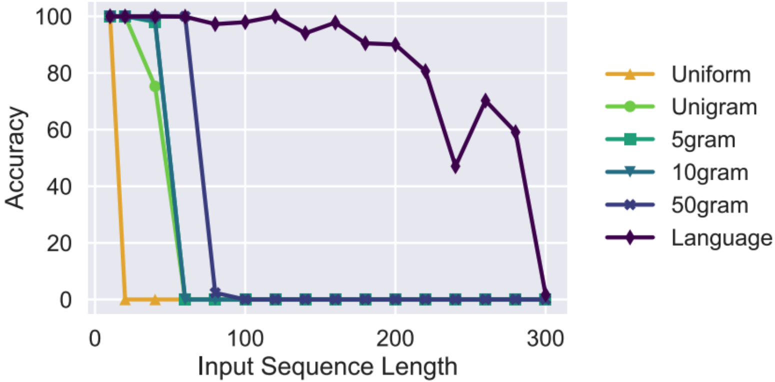
Liu, Levy, **Schwartz** et al., RepL4NLP 2018, best paper award



- Non-linguistic task

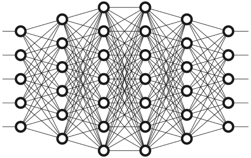
LSTMs Exploit Linguistic Attributes of Data

Liu, Levy, **Schwartz** et al., RepL4NLP 2018, best paper award

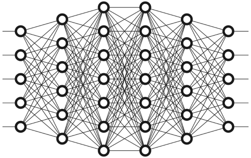


- Non-linguistic task
- Although they weren't designed that way, LSTMs do much better when trained on **language data**

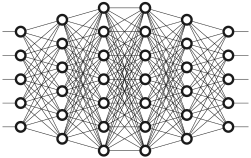
Case Study 2: Recurrent Neural Networks (RNN)

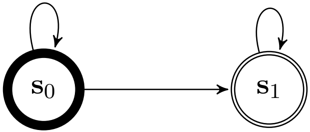


Interpretable
More robust
RNNs

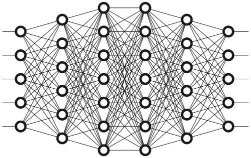


Case Study 2: Recurrent Neural Networks (RNN)

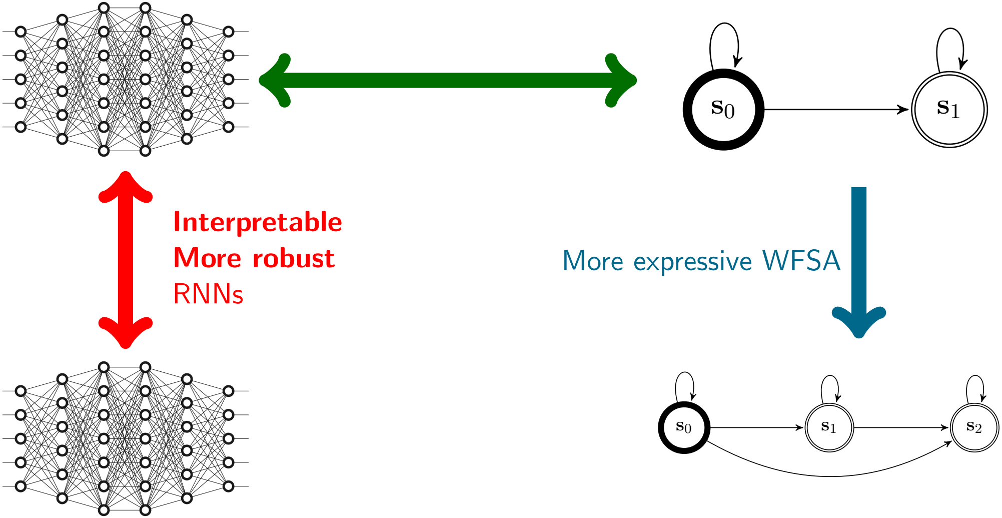




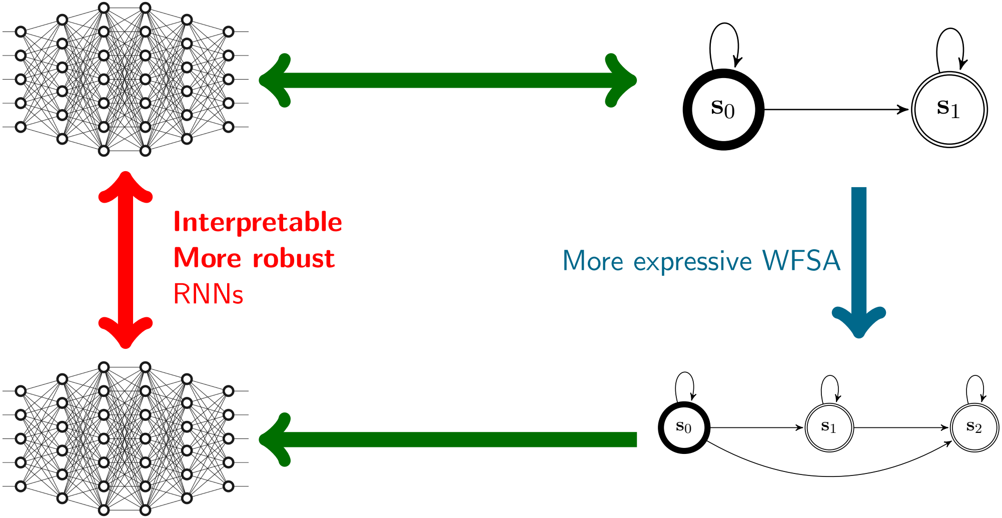
Interpretable
More robust
RNNs



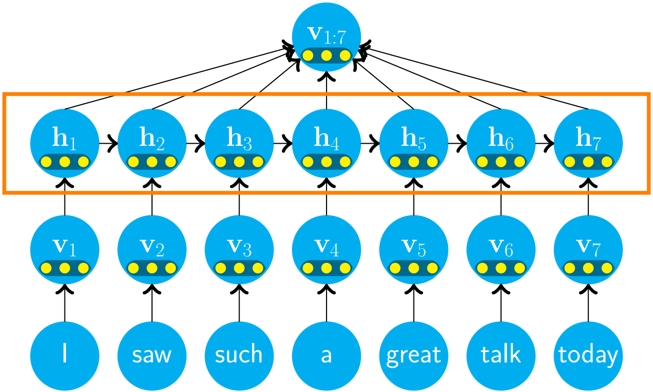
Case Study 2: Recurrent Neural Networks (RNN)



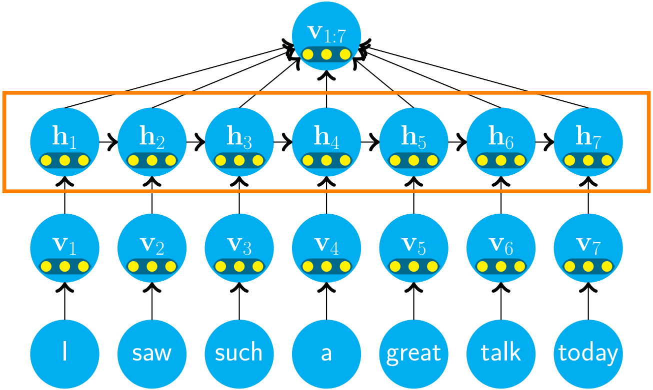
Case Study 2: Recurrent Neural Networks (RNN)



Recurrent Neural Networks: Hidden States



Recurrent Neural Networks: Hidden States



Recurrent function: $\boxed{\mathbf{h_i = f(h_{i-1}, v_i)}}$

Multiple Variants of Recurrent Neural Networks

- Elman (1990)
-  LSTM (Hochreiter and Schmidhuber, 1997)
- GRU (Cho et al., 2014)
- SGU (Gao and Glowacka, 2016)
- RAN (Lee et al., 2017)
- SCRN (Mikolov et al., 2014)
- T-RNN (Balduzzi and Ghifary, 2016)
- RCNN (Lei et al., 2016)
- Q-RNN (Bradbury et al., 2017)
- ISAN (Foerster et al., 2017)
- SoPa (Schwartz et al., 2018)
- SRU (Lei et al., 2018)

Multiple Variants of Recurrent Neural Networks

- Elman (1990)
-  LSTM (Hochreiter and Schmidhuber, 1997)
- GRU (Cho et al., 2014)
- SGU (Gao and Glowacka, 2016)
- RAN (Lee et al., 2017)
- SCRN (Mikolov et al., 2014)
- T-RNN (Balduzzi and Ghifary, 2016)
- RCNN (Lei et al., 2016)
- Q-RNN (Bradbury et al., 2017)
- ISAN (Foerster et al., 2017)
- SoPa (Schwartz et al., 2018)
- SRU (Lei et al., 2018)

- *What do different RNN variants have in common?*
- *What are they learning?*
- *Can we improve them?*

Example: Strongly-Typed Recurrent Neural Networks

Balduzzi and Ghifary (2016)

- A simple, competitive RNN
  - Draws inspiration from physics and functional programming
- $\mathbf{h}_i = \mathbf{z}_i \cdot \mathbf{h}_{i-1} + \mathbf{u}_i$
  - $\mathbf{z}_i, \mathbf{u}_i$ are non-linear parameterized functions of $\mathbf{v}_i$

Example: Strongly-Typed Recurrent Neural Networks

Balduzzi and Ghifary (2016)

- A simple, competitive RNN
  - Draws inspiration from physics and functional programming
- $\mathbf{h}_i = \mathbf{z}_i \cdot \mathbf{h}_{i-1} + \mathbf{u}_i$
  - $\mathbf{z}_i, \mathbf{u}_i$ are non-linear parameterized functions of $\mathbf{v}_i$
- Let $\mathbf{x}_i = [\mathbf{x}_i]_k$:

$$\mathbf{h}_n = \mathbf{z}_n \cdot \mathbf{h}_{n-1} + \mathbf{u}_n$$

Example: Strongly-Typed Recurrent Neural Networks

Balduzzi and Ghifary (2016)

- A simple, competitive RNN
  - Draws inspiration from physics and functional programming
- $\mathbf{h}_i = \mathbf{z}_i \cdot \mathbf{h}_{i-1} + \mathbf{u}_i$
  - $\mathbf{z}_i, \mathbf{u}_i$ are non-linear parameterized functions of $\mathbf{v}_i$
- Let $\mathbf{x}_i = [\mathbf{x}_i]_k$:

$$\begin{aligned}
\mathbf{h}_n &= \mathbf{z}_n \cdot \mathbf{h}_{n-1} + \mathbf{u}_n \\
&= \mathbf{z}_n \cdot (\mathbf{z}_{n-1} \cdot \mathbf{h}_{n-2} + \mathbf{u}_{n-1}) + \mathbf{u}_n
\end{aligned}$$

Example: Strongly-Typed Recurrent Neural Networks

Balduzzi and Ghifary (2016)

- A simple, competitive RNN
  - Draws inspiration from physics and functional programming
- $\mathbf{h}_i = \mathbf{z}_i \cdot \mathbf{h}_{i-1} + \mathbf{u}_i$
  - $\mathbf{z}_i, \mathbf{u}_i$ are non-linear parameterized functions of $\mathbf{v}_i$
- Let $\mathbf{x}_i = [\mathbf{x}_i]_k$:

$$
\begin{aligned}
\mathbf{h}_n &= \mathbf{z}_n \cdot \mathbf{h}_{n-1} + \mathbf{u}_n \\
&= \mathbf{z}_n \cdot (\mathbf{z}_{n-1} \cdot \mathbf{h}_{n-2} + \mathbf{u}_{n-1}) + \mathbf{u}_n \\
&= \mathbf{z}_n \cdot (\mathbf{z}_{n-1} \cdot (\mathbf{z}_{n-2} \cdot \mathbf{h}_{n-3} + \mathbf{u}_{n-2}) + \mathbf{u}_{n-1}) + \mathbf{u}_n
\end{aligned}
$$

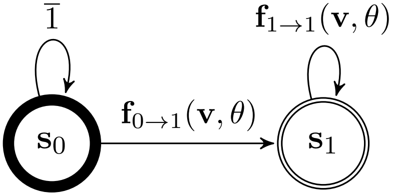
Example: Strongly-Typed Recurrent Neural Networks

Balduzzi and Ghifary (2016)

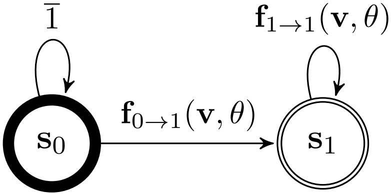
- A simple, competitive RNN
  - Draws inspiration from physics and functional programming
- $\mathbf{h}_i = \mathbf{z}_i \cdot \mathbf{h}_{i-1} + \mathbf{u}_i$
  - $\mathbf{z}_i, \mathbf{u}_i$ are non-linear parameterized functions of $\mathbf{v}_i$
- Let $\mathbf{x}_i = [\mathbf{x}_i]_k$:

$$
\begin{aligned}
\mathbf{h}_n &= \mathbf{z}_n \cdot \mathbf{h}_{n-1} + \mathbf{u}_n \\
&= \mathbf{z}_n \cdot (\mathbf{z}_{n-1} \cdot \mathbf{h}_{n-2} + \mathbf{u}_{n-1}) + \mathbf{u}_n \\
&= \mathbf{z}_n \cdot (\mathbf{z}_{n-1} \cdot (\mathbf{z}_{n-2} \cdot \mathbf{h}_{n-3} + \mathbf{u}_{n-2}) + \mathbf{u}_{n-1}) + \mathbf{u}_n \\
&= \dots \\
&= \sum_{i=1}^{n-1} \left(\mathbf{u}_i \prod_{j=i+1}^{n} \mathbf{z}_j \right) + \mathbf{u}_n
\end{aligned}
$$

Weighted Finite-State Automata!

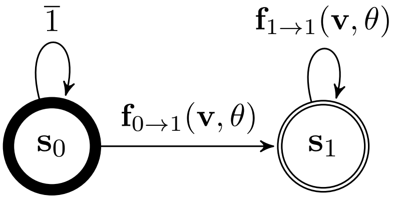


Weighted Finite-State Automata!



- Soft Pattern: W
  - Ignore the self-loops for simplicity

Weighted Finite-State Automata!



- Soft Pattern: W
  - Ignore the self-loops for simplicity
- $\mathcal{S}_2\ (\mathbf{v}_1 : \mathbf{v}_n) = \sum_{i=1}^{n-1} \left(\mathbf{f}_{0\to 1}(\mathbf{v}_i,\theta) \prod_{j=i+1}^{n} \mathbf{f}_{1\to 1}(\mathbf{v}_j,\theta) \right) + \mathbf{f}_{0\to 1}(\mathbf{v}_n,\theta)$

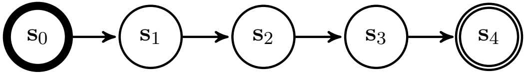
Strongly-Typed RNNs are Rational!

Can Be Computed Using a Set of WFSAs

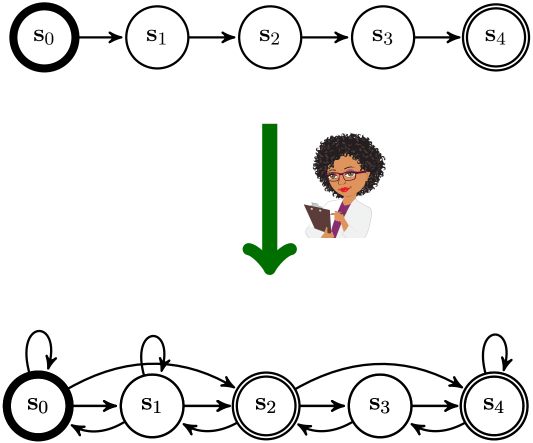
$$\mathbf{h}_n = \sum_{i=1}^{n-1} \left(\mathbf{u}_i \prod_{j=i+1}^{n} \mathbf{z}_j \right) + \mathbf{u}_n$$

$$\mathcal{S}_2(\mathbf{v}_1 : \mathbf{v}_n) = \sum_{i=1}^{n-1} \left(\mathbf{f}_{0 \to 1}(\mathbf{v}_i, \theta) \prod_{j=i+1}^{n} \mathbf{f}_{1 \to 1}(\mathbf{v}_j, \theta) \right) + \mathbf{f}_{0 \to 1}(\mathbf{v}_n, \theta)$$

Work in Progress 3: Make your own Deep Model!



Work in Progress 3: Make your own Deep Model!



David Balduzzi and Muhammad Ghifary. 2016. Strongly-typed recurrent neural networks. In *Proc. of ICML*.

James Bradbury, Stephen Merity, Caiming Xiong, and Richard Socher. 2017. Quasi-recurrent neural network. In *Proc. of ICLR*.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. In *Proc. of SSST*.

Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science*, 14(2):179–211.

Jakob N. Foerster, Justin Gilmer, Jan Chorowski, Jascha Sohl-Dickstein, and David Sussillo. 2017. Intelligible language modeling with input switched affine networks. In *Proc. of ICML*.

Yuan Gao and Dorota Glowacka. 2016. Deep gate recurrent neural network. In *Proc. of ACML*, pages 350–365.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2017. Recurrent additive networks. arXiv:1705.07393.

Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Kateryna Tymoshenko, Alessandro Moschitti, and Lluís Màrquez. 2016. Semi-supervised question retrieval with gated convolutions. In *Proc. of NAACL*.

Tao Lei, Yu Zhang, Sida I. Wang, Hui Dai, and Yoav Artzi. 2018. Simple recurrent units for highly parallelizable recurrence. In *Proc. of EMNLP*.

Tomas Mikolov, Armand Joulin, Sumit Chopra, Michaël Mathieu, and Marc'Aurelio Ranzato. 2014. Learning longer memory in recurrent neural networks. arXiv:1412.7753.

Marcel Paul Schützenberger. 1961. On the definition of a family of automata. *Information and Control*, 4(2-3):245–270.

Roy Schwartz, Sam Thomson, and Noah A. Smith. 2018. SoPa: Bridging CNNs, RNNs, and weighted finite-state machines. In *Proc. of ACL*.

A. Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269.