Neural Models over Tree Structures

Mausam

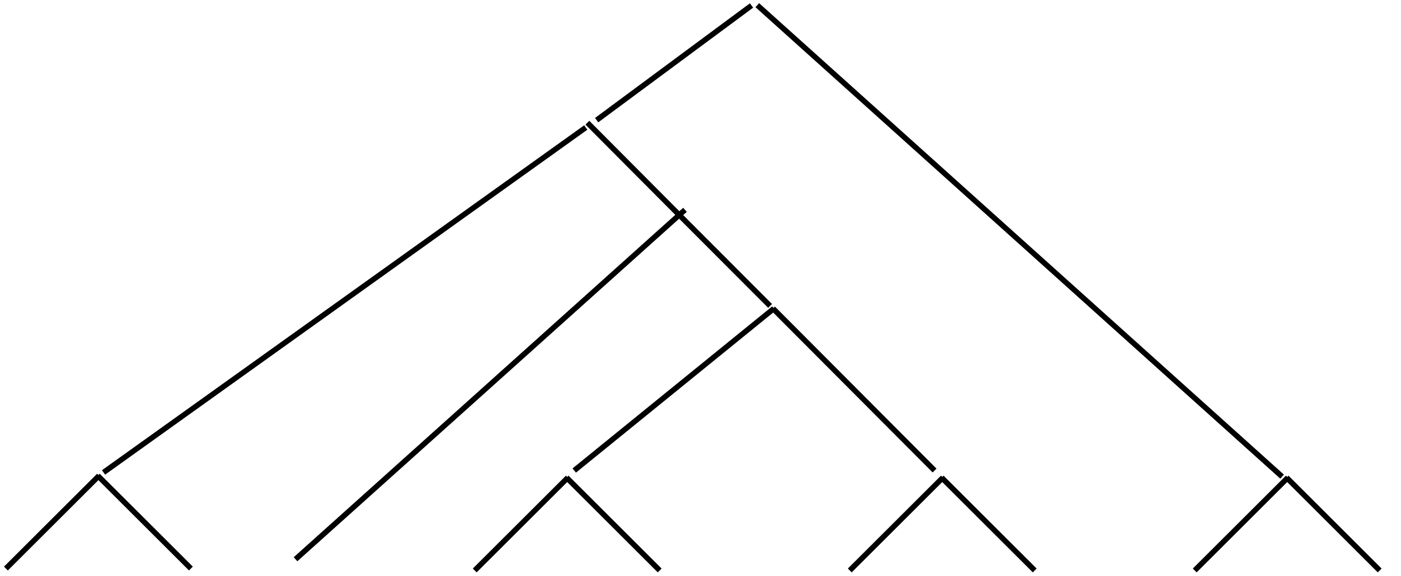
(Slides by Yoav Goldberg, Richard Socher, Daniel Perez)

Trees

- Sequences are nice.
- But when working with language, we often see tree structures.
- An RNN encodes a sequence as a vector.
- We would like to **encode a tree as a vector**

The boy who always wears blue shirts went home

(((The boy) (who (always wears) (blue shirts))) went home)



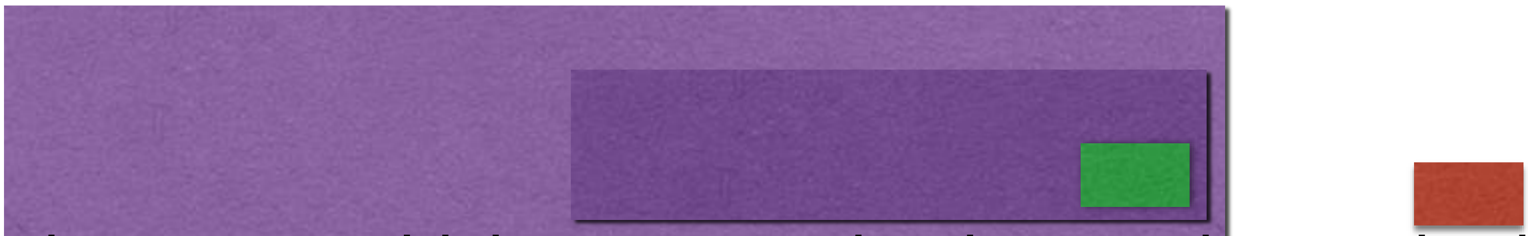
The boy who always wears blue shirts went home

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

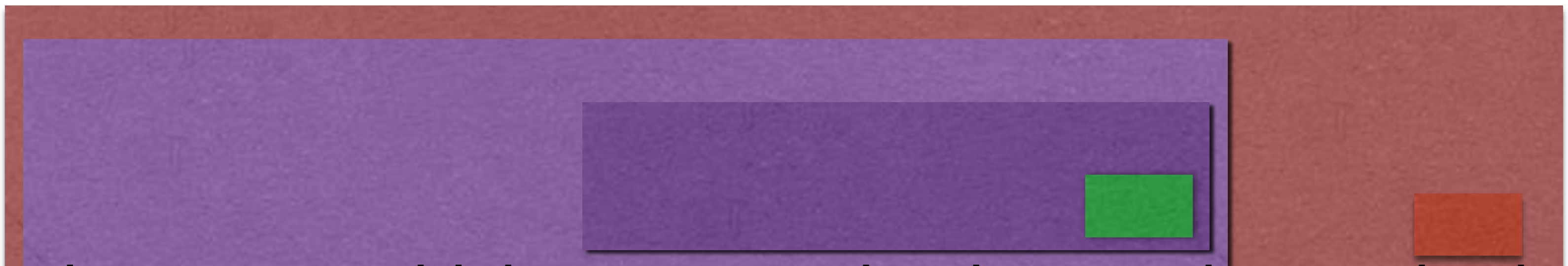
the soup , which I expected to be good , was bad

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

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



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

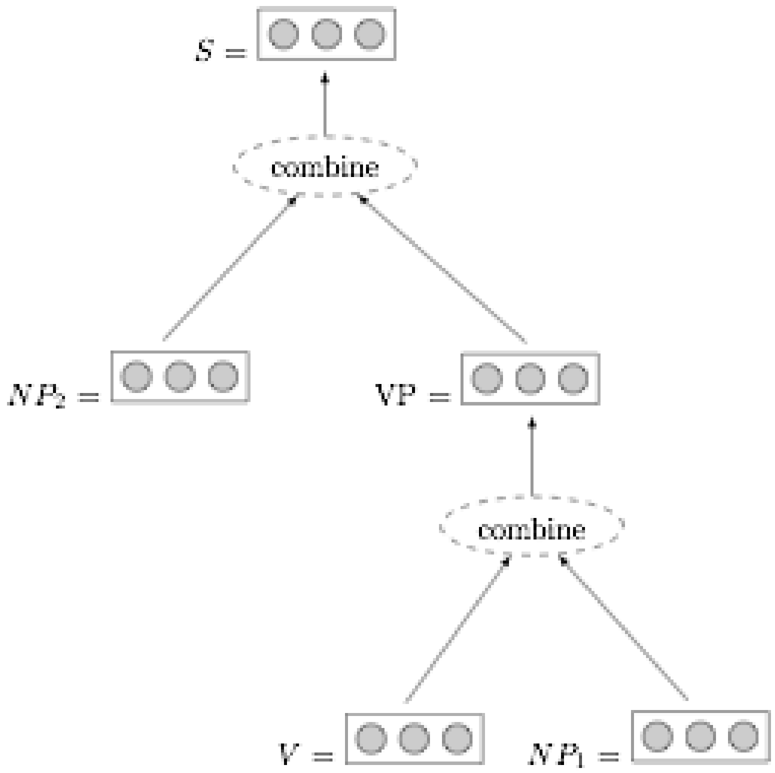


the soup , which I expected to be good , was bad

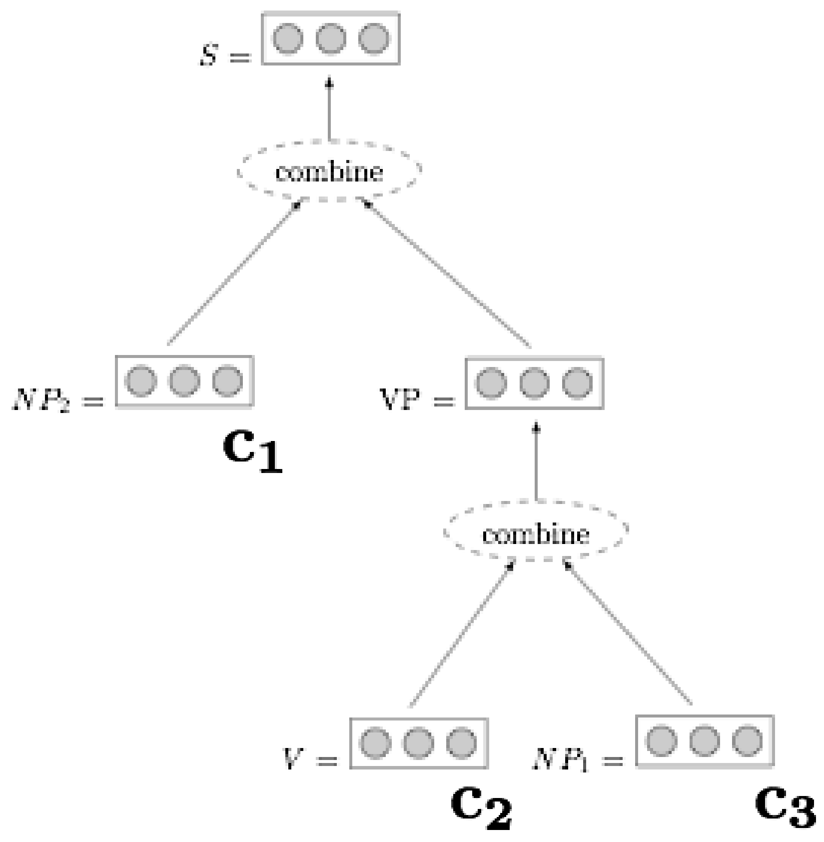
Trees

- Sequences are nice.
- But when working with language, we often see tree structures.
- An RNN encodes a sequence as a vector.
- We would like to **encode a tree as a vector**.

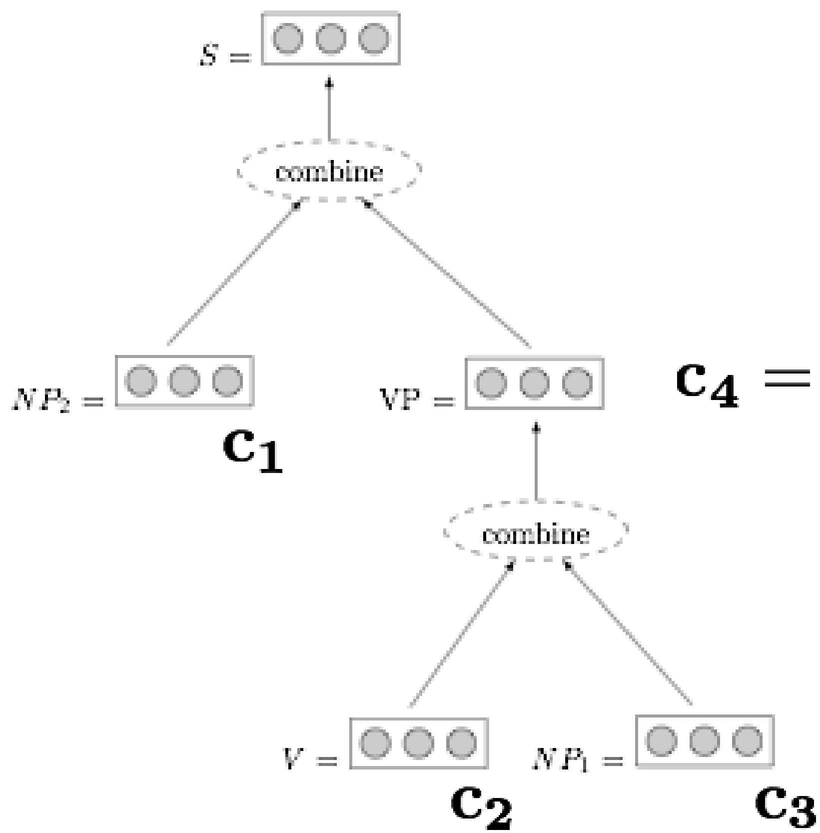
Recursive Neural Nets



Recursive Neural Nets

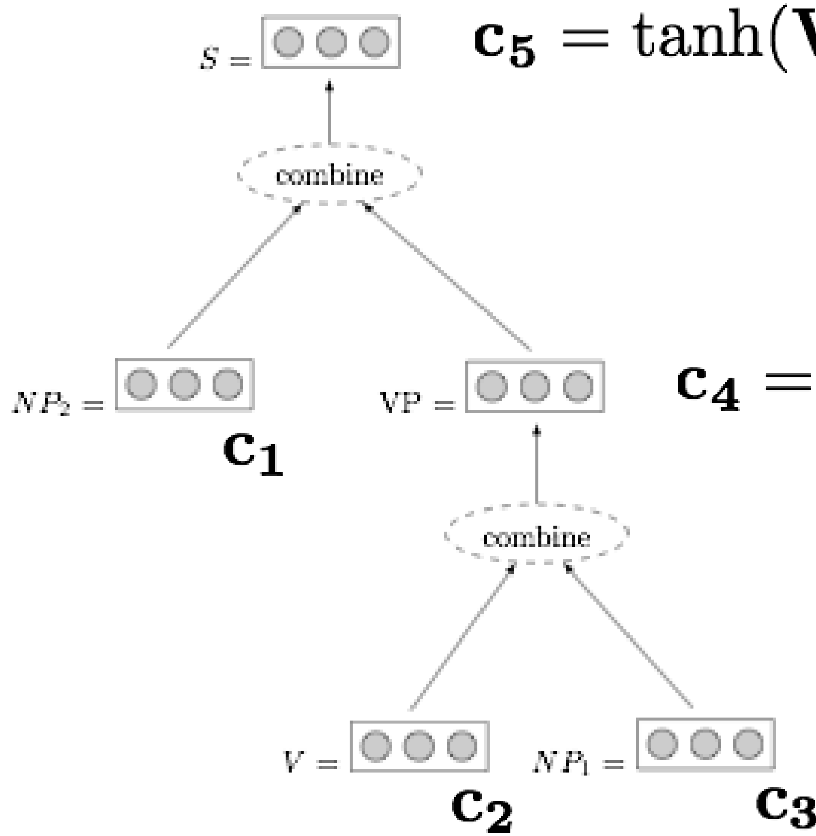


Recursive Neural Nets



$$\mathbf{c_4} = \tanh(\mathbf{W}[\mathbf{c_2}; \mathbf{c_3}])$$

Recursive Neural Nets



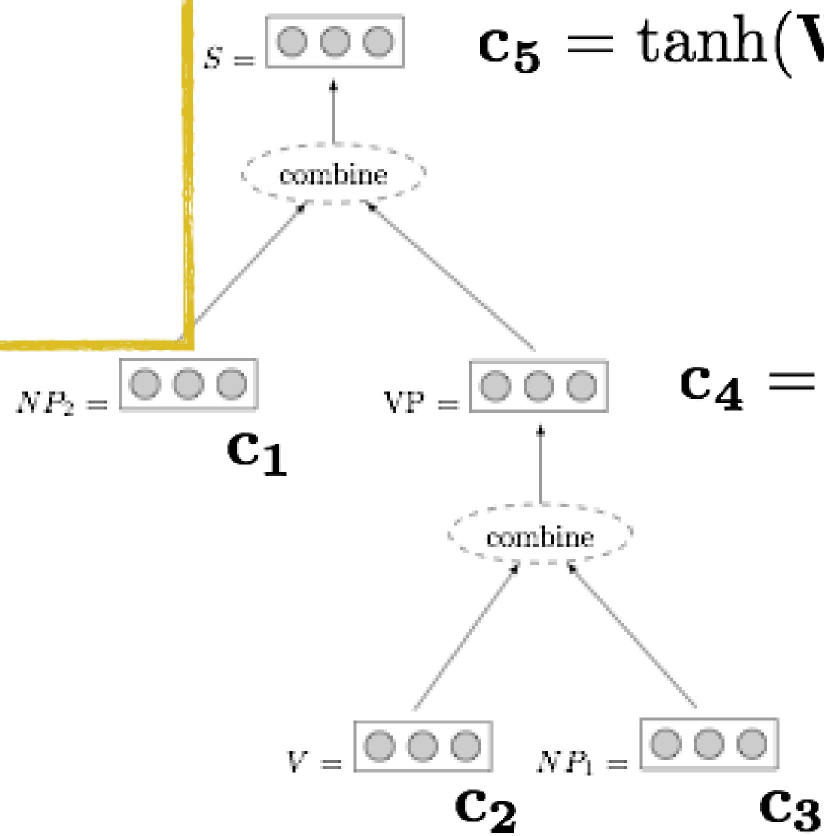
$$\mathbf{c_5} = \tanh(\mathbf{W}[\mathbf{c_1};\mathbf{c_4}])$$

$$\mathbf{c_4} = \tanh(\mathbf{W}[\mathbf{c_2};\mathbf{c_3}])$$

Recursive Neural Nets

Like recurrent nets:
we have **shared parameters** at nodes.
but structure is
no longer a sequence.

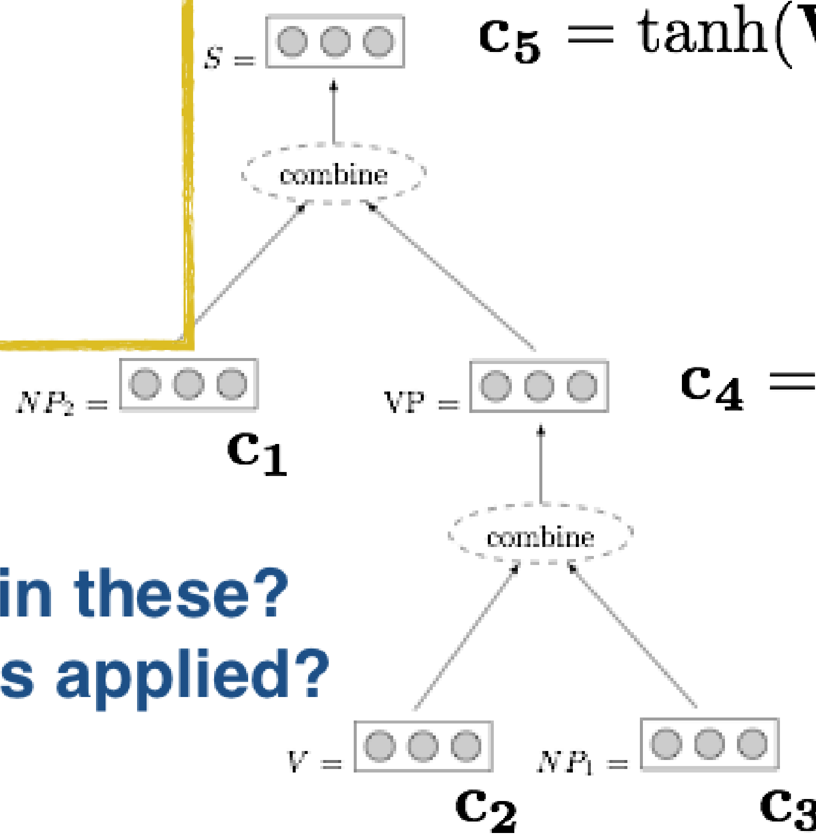
$$\mathbf{c_5} = \tanh(\mathbf{W}[\mathbf{c_1}; \mathbf{c_4}])$$



$$\mathbf{c_4} = \tanh(\mathbf{W}[\mathbf{c_2}; \mathbf{c_3}])$$

Recursive Neural Nets

Like recurrent nets:
we have **shared parameters** at nodes.
but structure is
no longer a sequence.

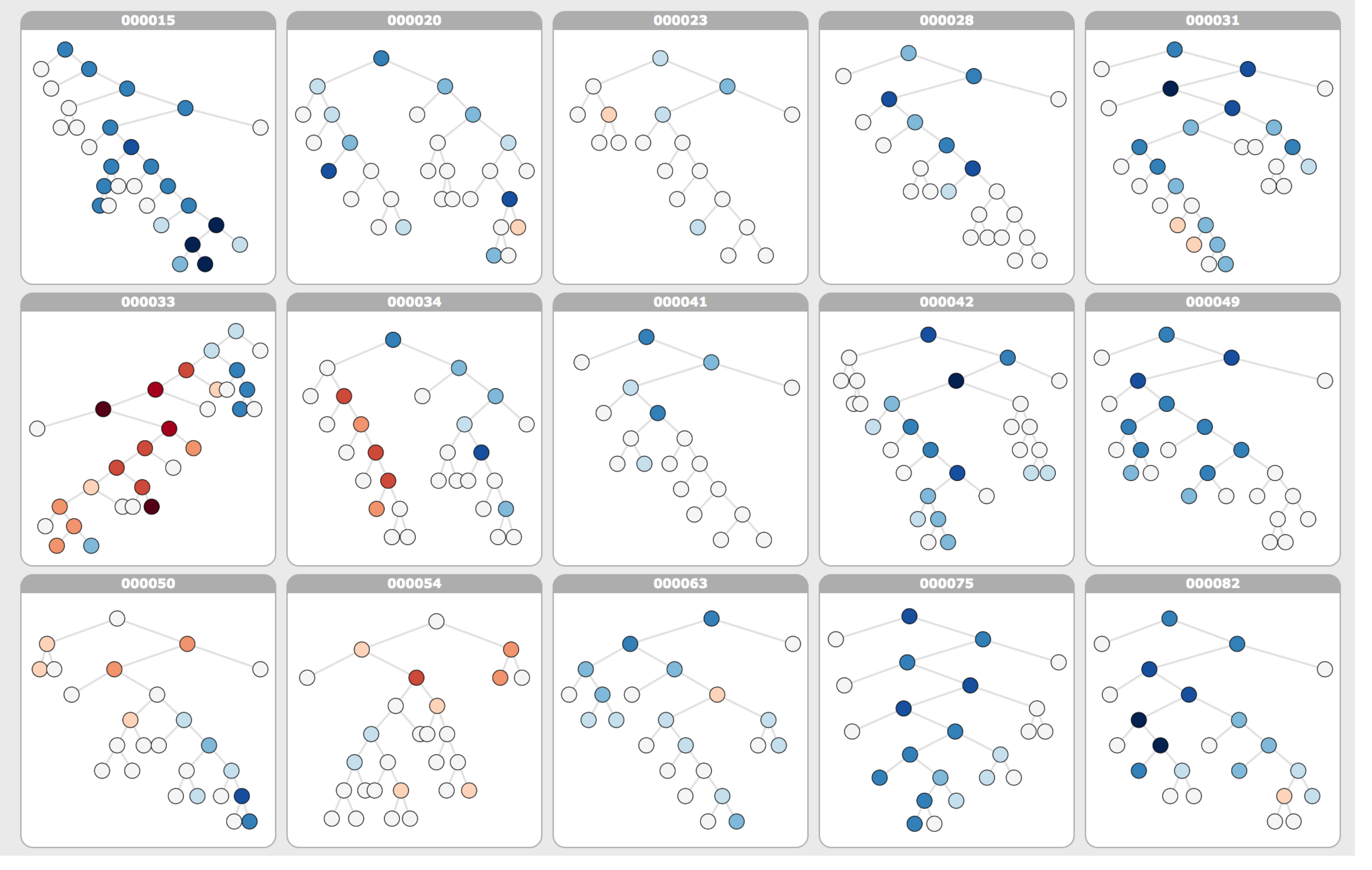


$$\mathbf{c}_5 = \tanh(\mathbf{W}[\mathbf{c}_1; \mathbf{c}_4])$$

$$\mathbf{c}_4 = \tanh(\mathbf{W}[\mathbf{c}_2; \mathbf{c}_3])$$

How do we train these?
where is the loss applied?

Stanford Sentiment Treebank

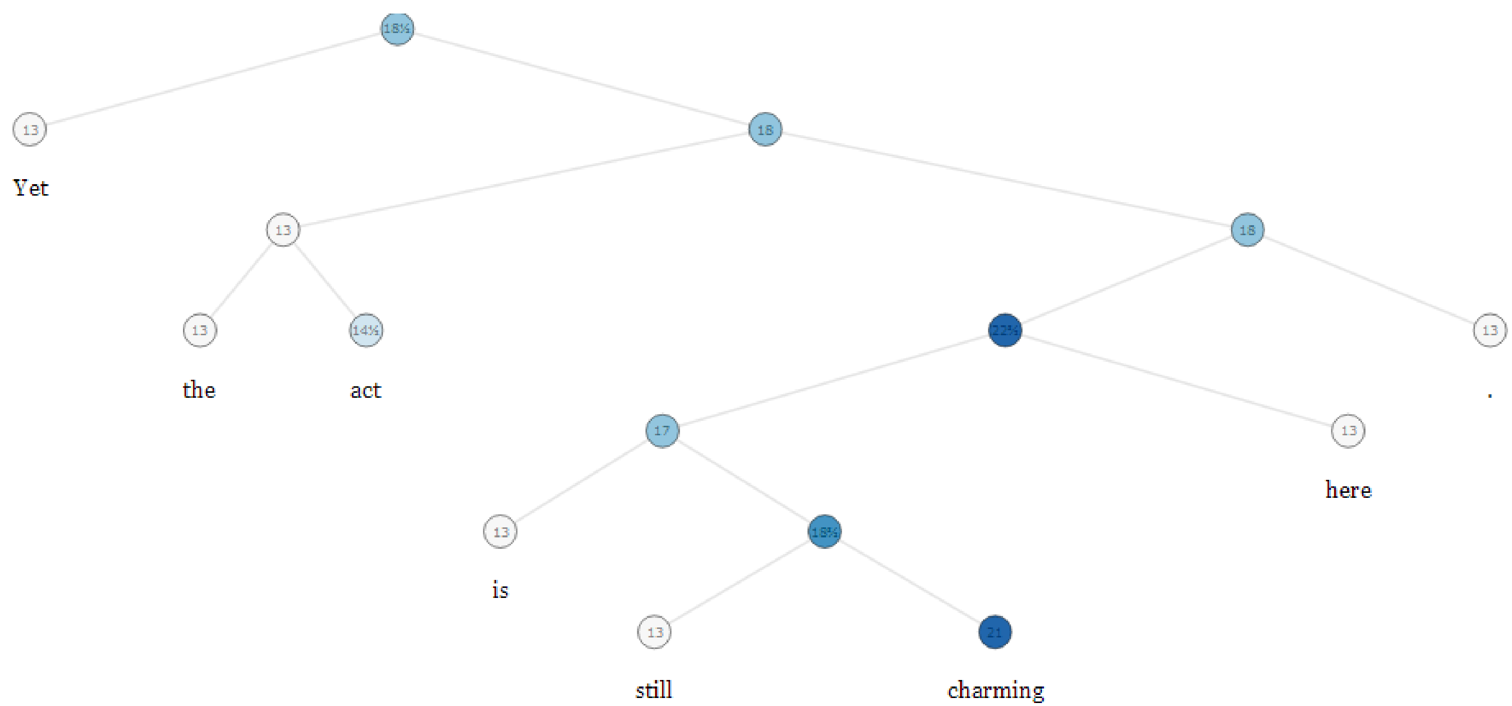


Need for a Sentiment Treebank

- Almost all work on sentiment analysis has used mostly word-order independent methods
- But many papers acknowledge that sentiment interacts with syntax in complex ways
- Little work has been done on these interactions because they're very difficult to learn
- Single-sentence sentiment classification accuracy has languished at ~80% for a long time

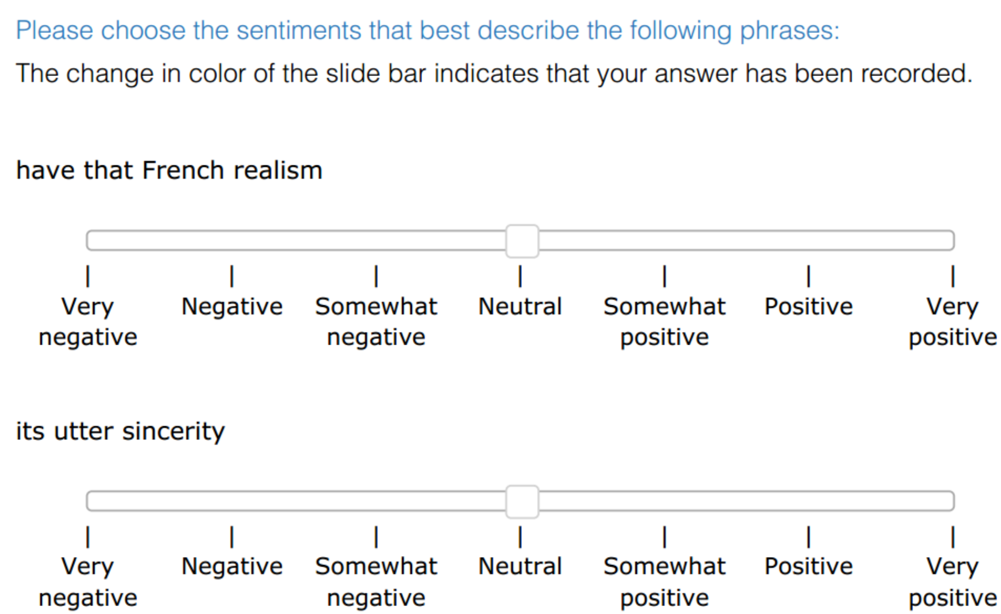
Goal of the Sentiment Treebank

- At every level of the parse tree, annotate the sentiment of the phrase it subsumes
- Use a 5-class scheme (--, -, 0, +, ++)

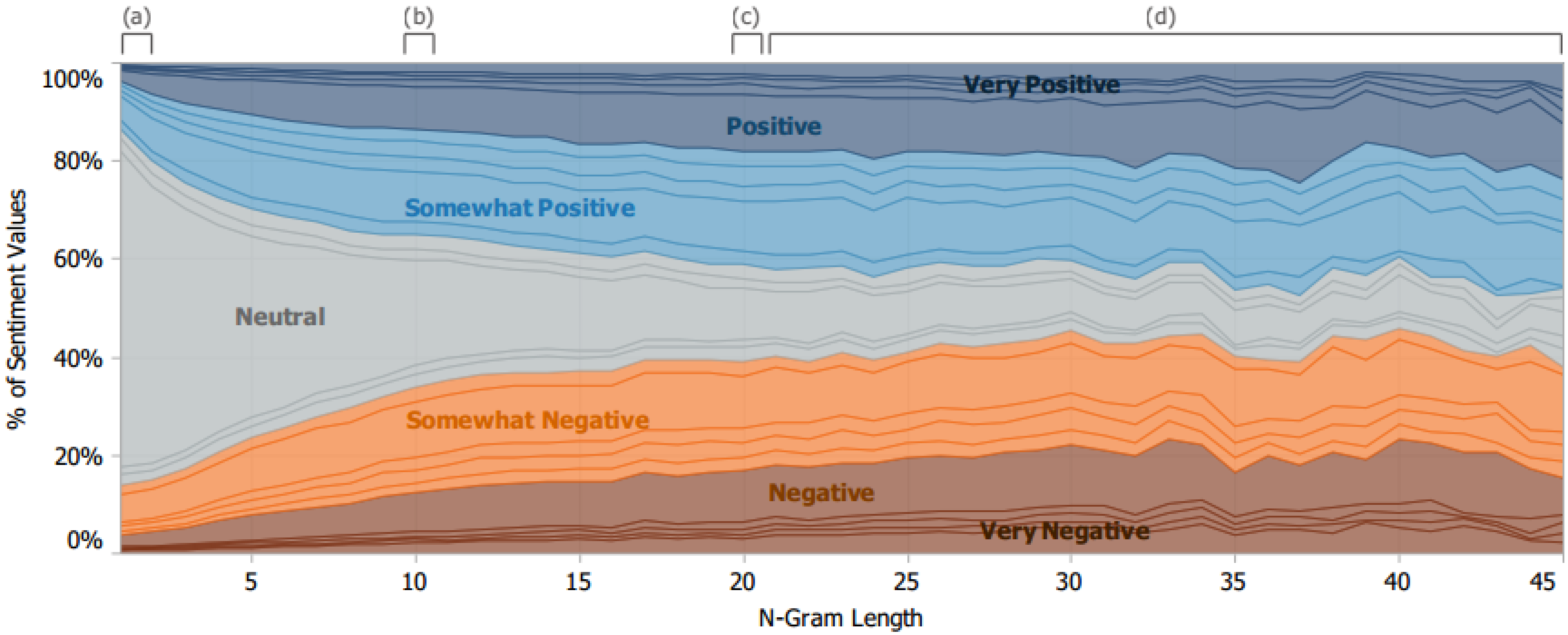


Construction of the Sentiment Treebank

- For 11,855 sentences, parse and break into phrases (215,154 total)
- The sentiment of each phrase is annotated with Mechanical Turk



Construction of the Sentiment Treebank

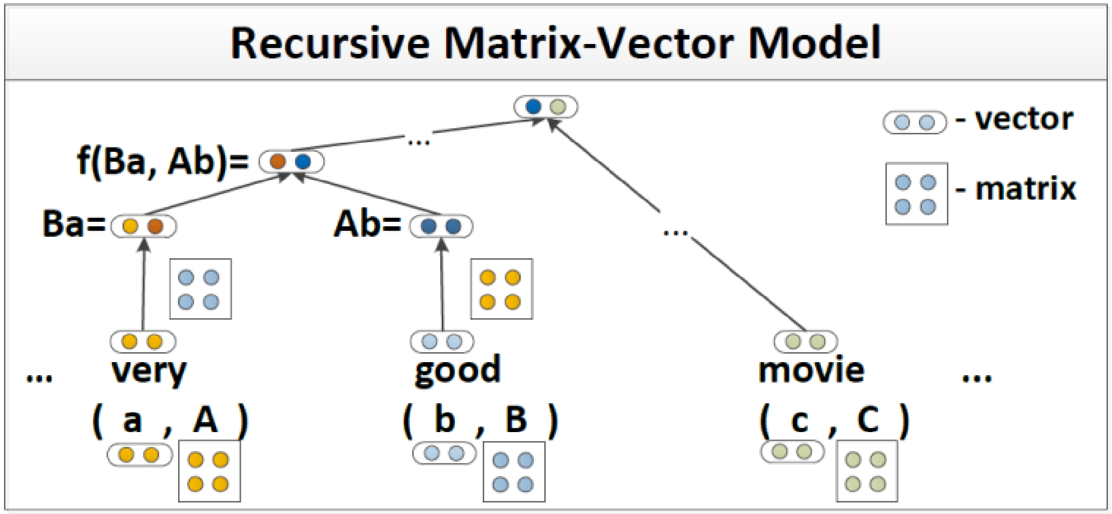


Matrix Vector RNN (MV-RNN)

- Each word has both
  - An associated vector (it's meaning)
  - An associated matrix (it's personal composition function)

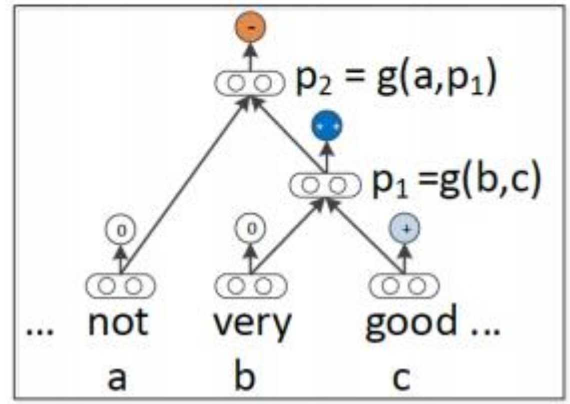
This is a good idea, but in practice, it's way too many parameters to learn

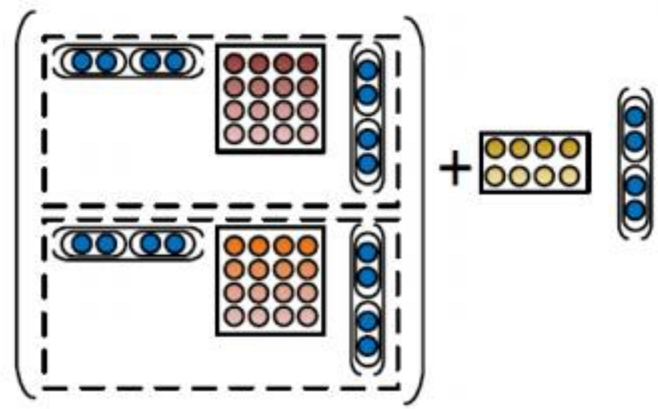
If the vectors are d-dimensional, then every word, has $(d+1) \times d$ parameters.



Recursive Neural Tensor Network (RTNN)

- At a high level:
  - The composition function is a tensor, which means expressiveness, with fewer parameters to learn
  - In the same way that similar words have similar vectors, this lets similar words have similar composition behavior





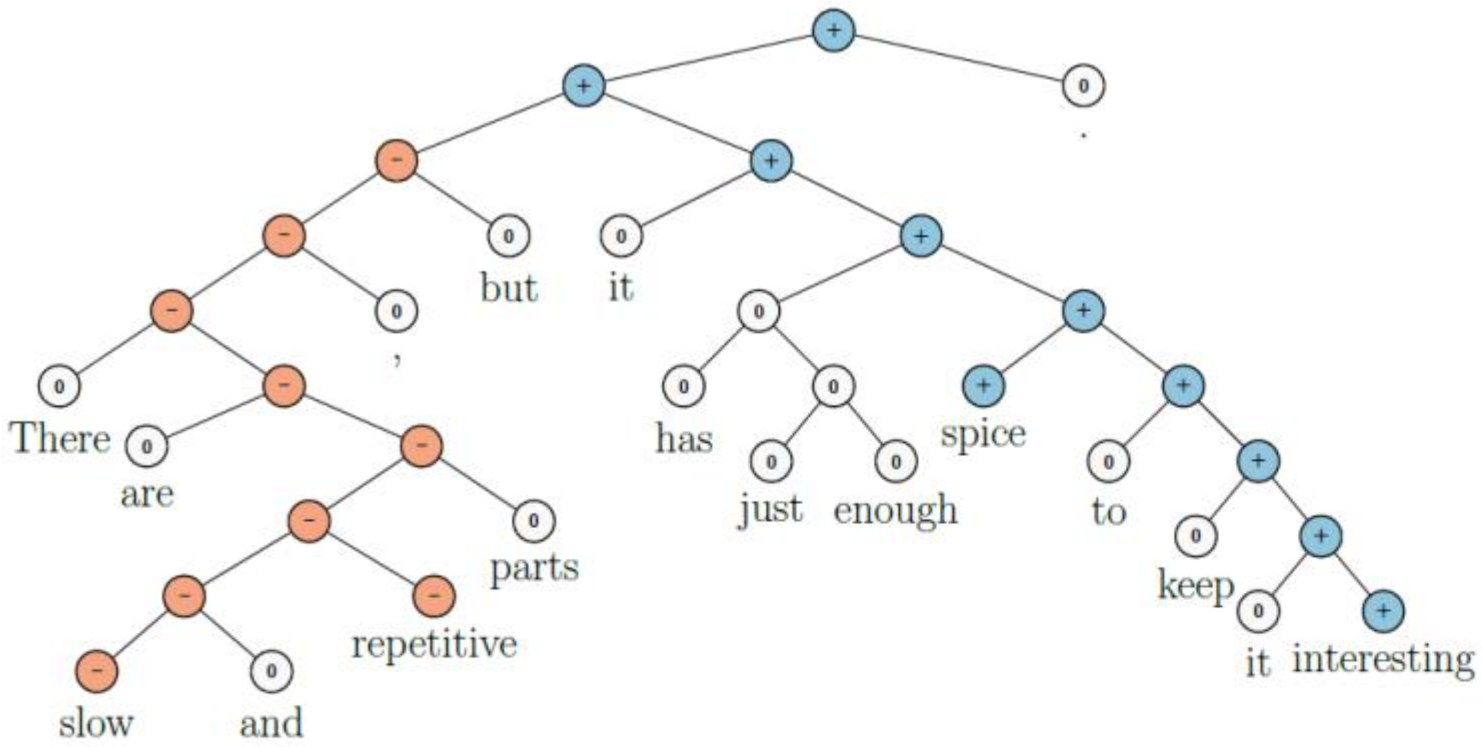
$$h = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix}; h_i = \begin{bmatrix} b \\ c \end{bmatrix}^T V^{[i]} \begin{bmatrix} b \\ c \end{bmatrix}.$$

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}\right)$$

$$p_2 = f\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} a \\ p_1 \end{bmatrix} + W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

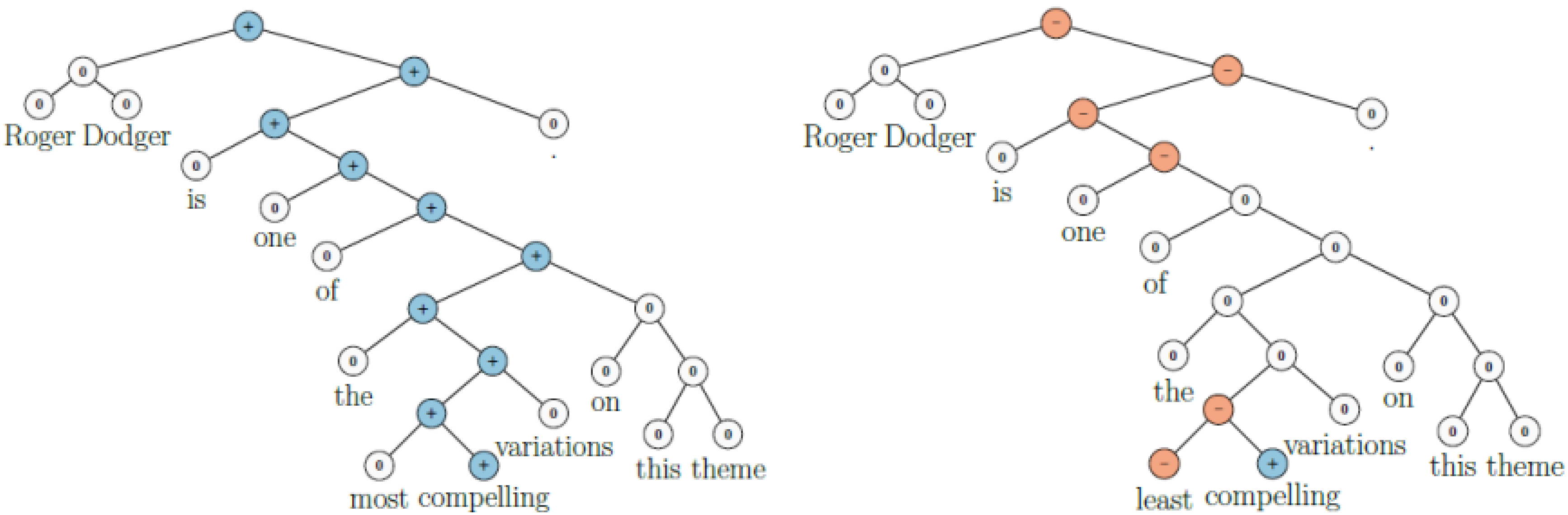
What is this model able to do?

- Learns structures like “X but Y”



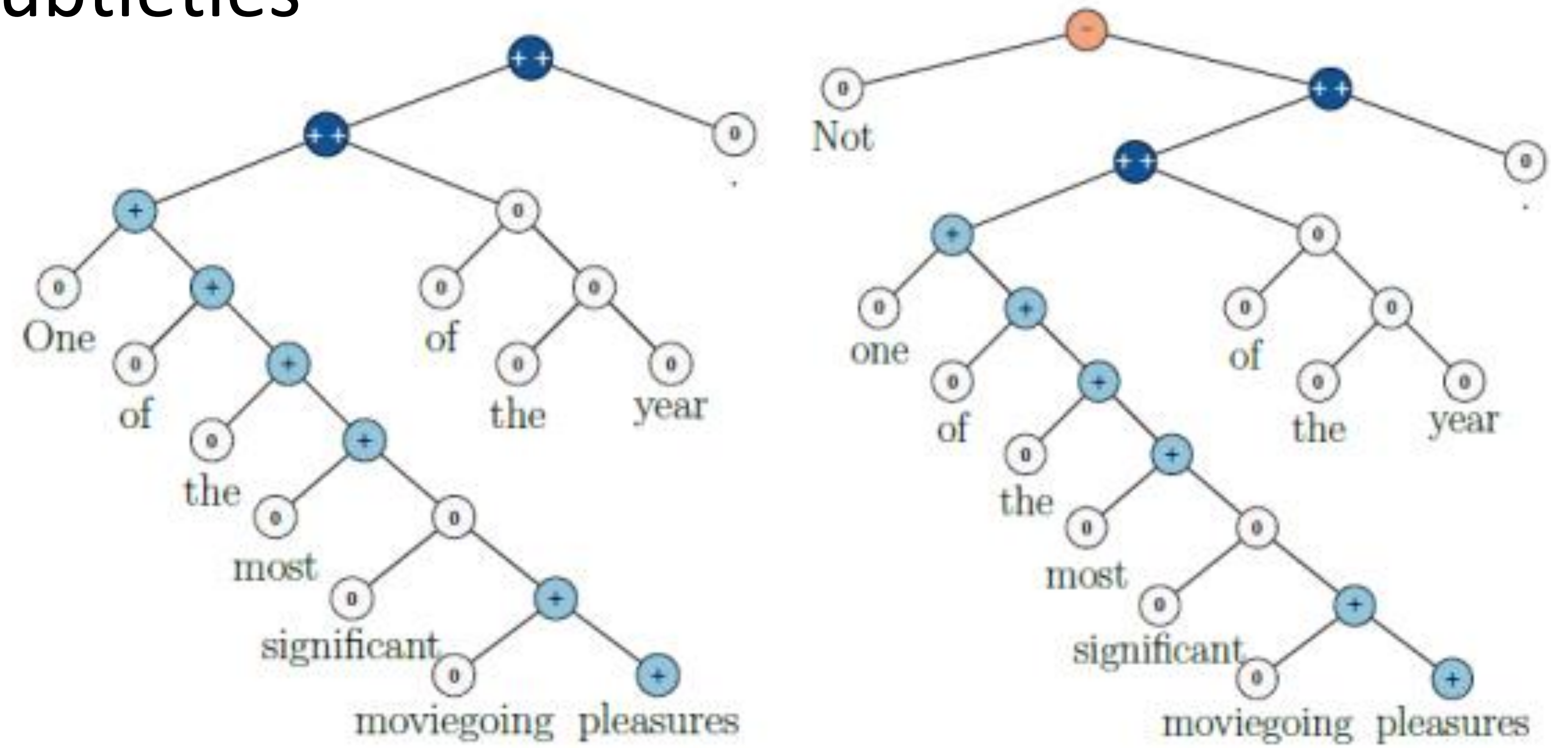
What is this model able to do?

- Small changes are able to propagate all the way up the tree

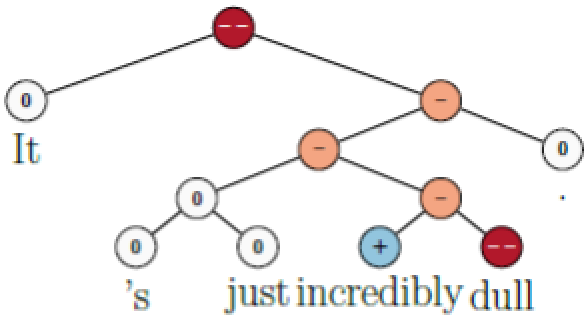


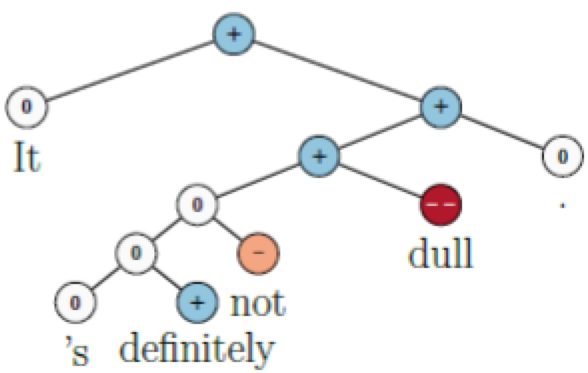
What is this model able to do?

- Learns how negation works, including many subtleties

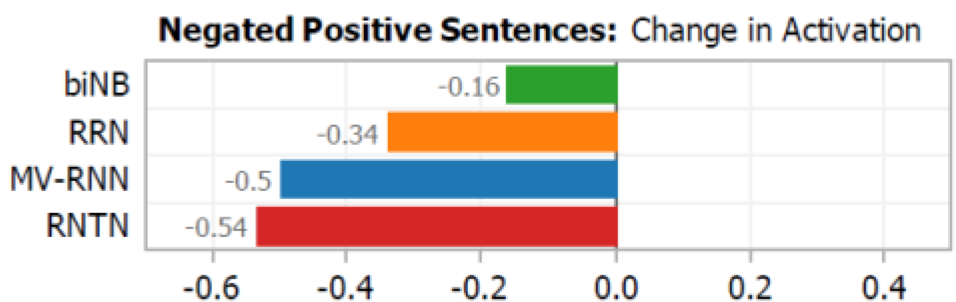


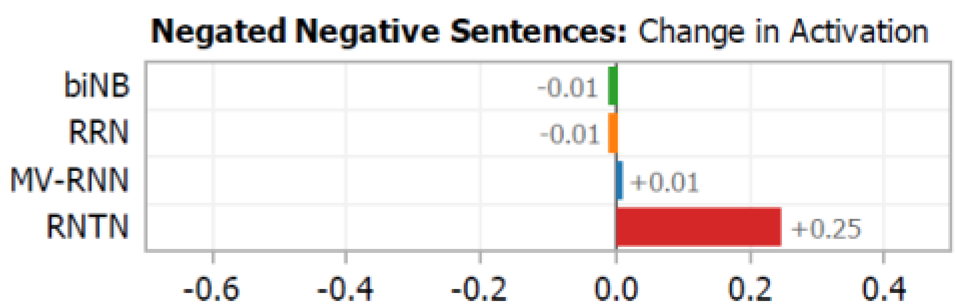
Negation Evaluation





Model	Accuracy	
	Negated Positive	Negated Negative
biNB	19.0	27.3
RNN	33.3	45.5
MV-RNN	52.4	54.6
RNTN	**71.4**	**90.9**

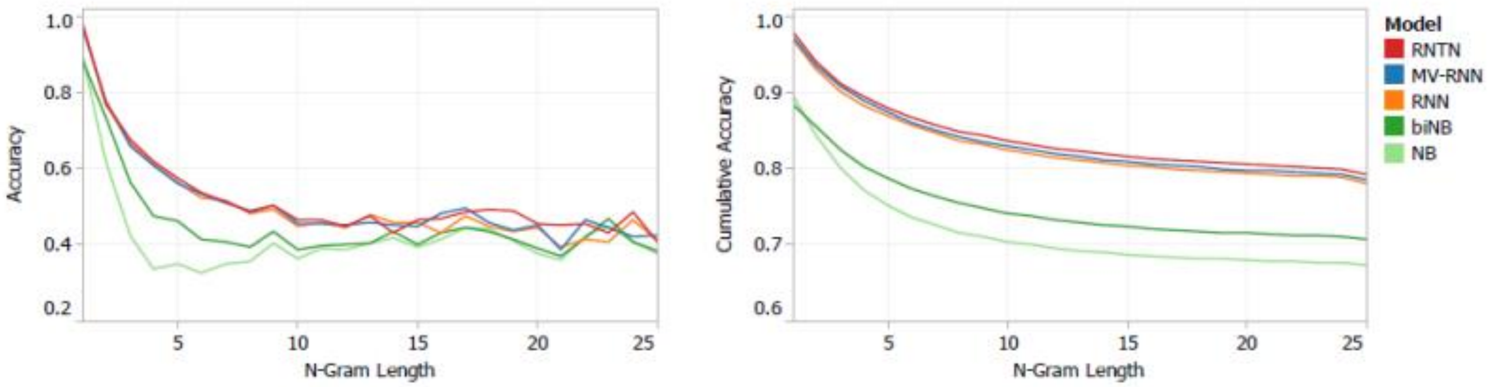




Positive and Negative N-grams

n	**Most positive n-grams**	**Most negative n-grams**
1	engaging ; best ; powerful ; love ; beautiful ; entertaining ; clever ; terrific ; excellent ; great ;	bad ; dull ; boring ; fails ; worst ; stupid ; painfully ; cheap ; forgettable ; disaster ;
2	excellent performances ; amazing performance ; terrific performances ; A masterpiece ; masterful film ; wonderful film ; terrific performance ; masterful piece ; wonderful movie ; marvelous performances ;	worst movie ; bad movie ; very bad ; shapeless mess ; worst thing ; tepid waste ; instantly forgettable ; bad film ; extremely bad ; complete failure ;
3	an amazing performance ; a terrific performance ; a wonderful film ; wonderful all-ages triumph ; A masterful film ; a wonderful movie ; a tremendous performance ; drawn excellent performances ; most visually stunning ; A stunning piece ;	for worst movie ; A lousy movie ; most joyless movie ; a complete failure ; another bad movie ; fairly terrible movie ; a bad movie ; extremely unfunny film ; most painfully marginal ; very bad sign ;
5	nicely acted and beautifully shot ; gorgeous imagery , effective performances ; the best of the year ; a terrific American sports movie ; very solid , very watchable ; a fine documentary does best ; refreshingly honest and ultimately touching ;	silliest and most incoherent movie ; completely crass and forgettable movie ; just another bad movie . ; drowns out the lousy dialogue ; a fairly terrible movie ... ; A cumbersome and cliche-ridden movie ; a humorless , disjointed mess ;
8	one of the best films of the year ; simply the best family film of the year ; the best film of the year so far ; A love for films shines through each frame ; created a masterful piece of artistry right here ; A masterful film from a master filmmaker , ; 's easily his finest American film ... comes ;	A trashy , exploitative , thoroughly unpleasant experience ; this sloppy drama is an empty vessel . ; a meandering , inarticulate and ultimately disappointing film ; an unimaginative , nasty , glibly cynical piece ; bad , he 's really bad , and ; quickly drags on becoming boring and predictable . ; be the worst special-effects creation of the year ;

Sentiment Analysis Evaluation



Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.6**	**87.6**	**85.4**

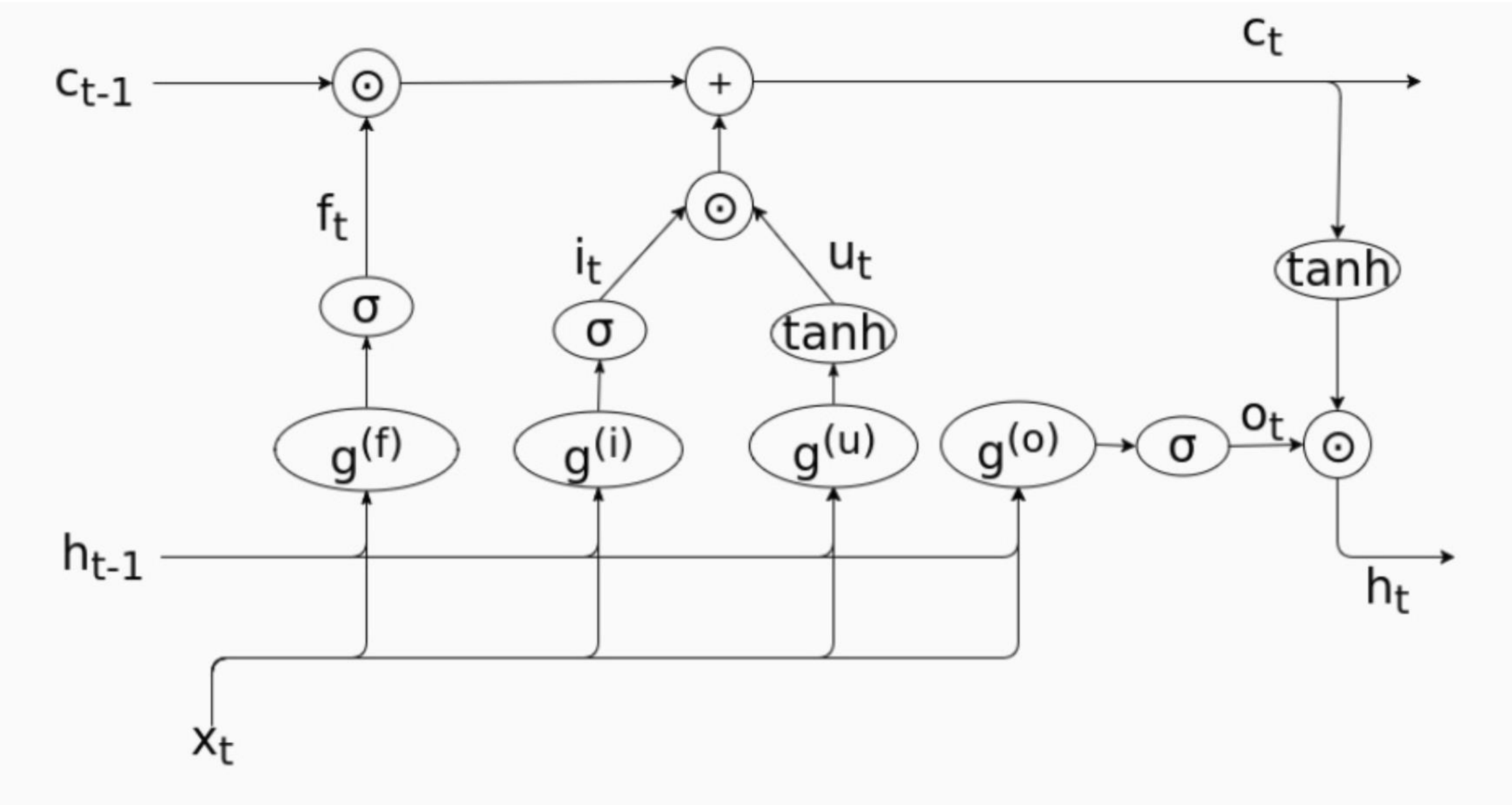
Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng and Christopher Potts
Stanford University, Stanford, CA 94305, USA
richard@socher.org, {aperelyg, jcchuang, ang}@cs.stanford.edu
{jeaneis, manning, cgpotts}@stanford.edu

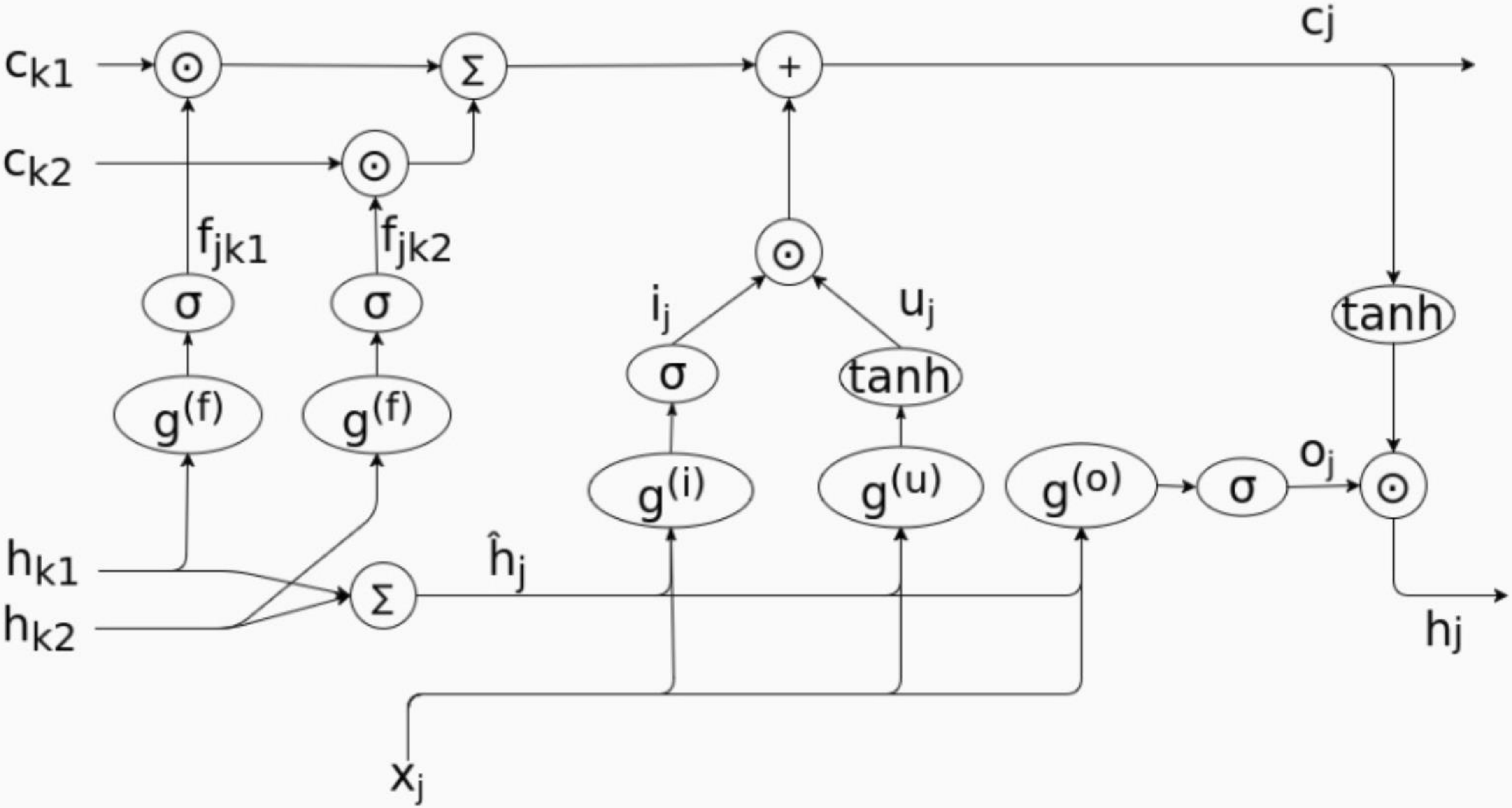
LSTM RNN

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

LSTM



Child Sum Tree LSTM



Child-sum tree LSTM at node j with children k_1 and k_2

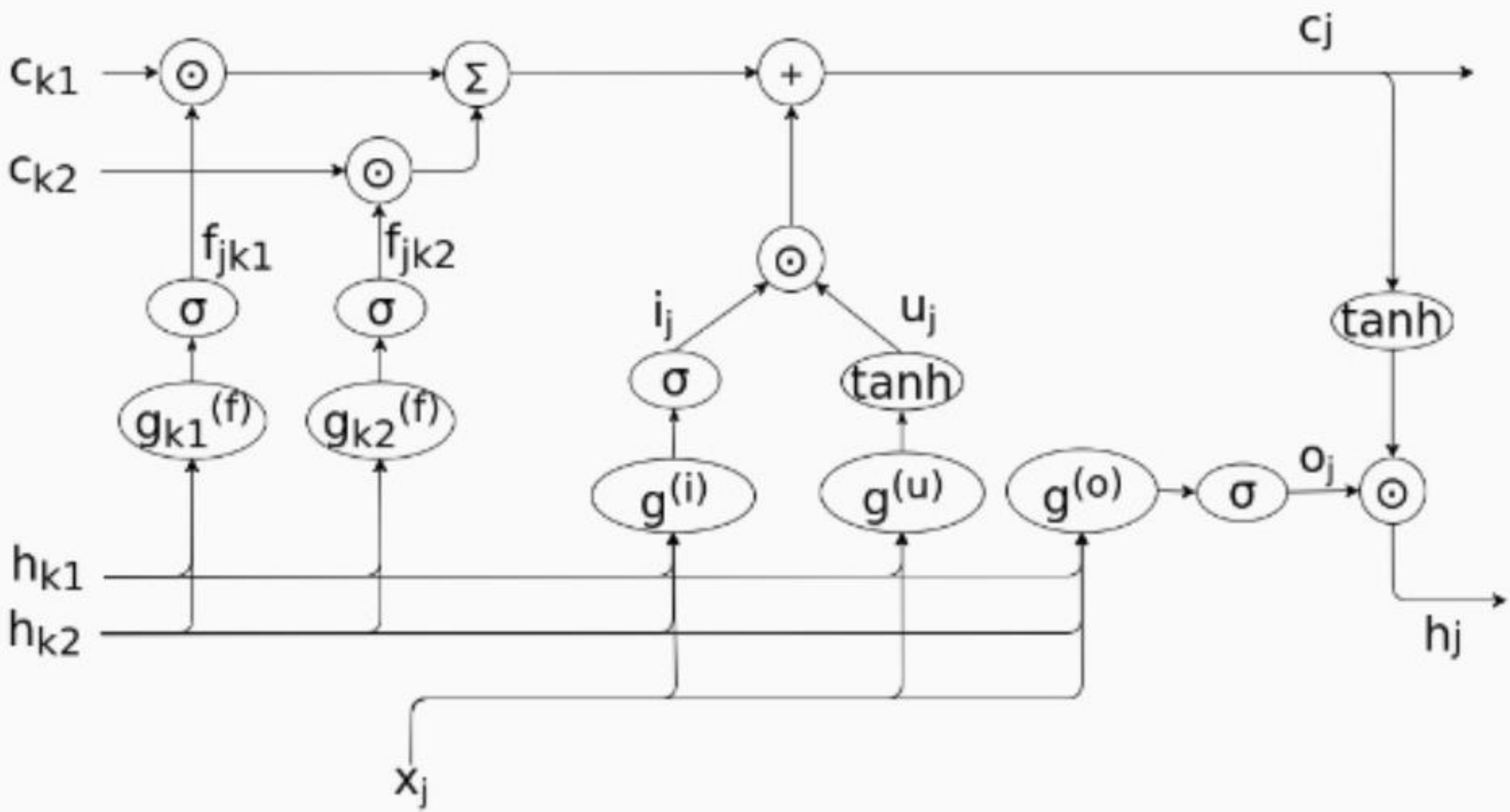
$$
\begin{aligned}
\tilde{h}_j &= \sum_{k \in C(j)} h_k, \\
i_j &= \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right), \\
f_{jk} &= \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right), \\
o_j &= \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right), \\
u_j &= \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right), \\
c_j &= i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k, \\
h_j &= o_j \odot \tanh(c_j),
\end{aligned}
$$

Child Sum Tree LSTM

- does not take into account child order
- works with variable number of children
  – good for dependency parses
- shares gates weight among children

- Application
  – Dependency tree LSTM

N-ary Tree LSTM



Binary tree LSTM at node j with children k_1 and k_2

$$i_j = \sigma\left(W^{(i)}x_j + \sum_{\ell=1}^{N} U_\ell^{(i)} h_{j\ell} + b^{(i)}\right),$$

$$f_{jk} = \sigma\left(W^{(f)}x_j + \sum_{\ell=1}^{N} U_{k\ell}^{(f)} h_{j\ell} + b^{(f)}\right),$$

$$o_j = \sigma\left(W^{(o)}x_j + \sum_{\ell=1}^{N} U_\ell^{(o)} h_{j\ell} + b^{(o)}\right),$$

$$u_j = \tanh\left(W^{(u)}x_j + \sum_{\ell=1}^{N} U_\ell^{(u)} h_{j\ell} + b^{(u)}\right)$$

$$c_j = i_j \odot u_j + \sum_{\ell=1}^{N} f_{j\ell} \odot c_{j\ell},$$

$$h_j = o_j \odot \tanh(c_j),$$

N-ary Tree LSTM

- Each node must have at most N children
- Fine-grained control on how information propagates
- Forget gate parameterized such that siblings can affect the computation

- Application
  - Constituency Tree LSTM

Sentiment Treebank Results

Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks

Kai Sheng Tai, Richard Socher*, Christopher D. Manning
Computer Science Department, Stanford University, *MetaMind Inc.
kst@cs.stanford.edu, richard@metamind.io, manning@stanfor

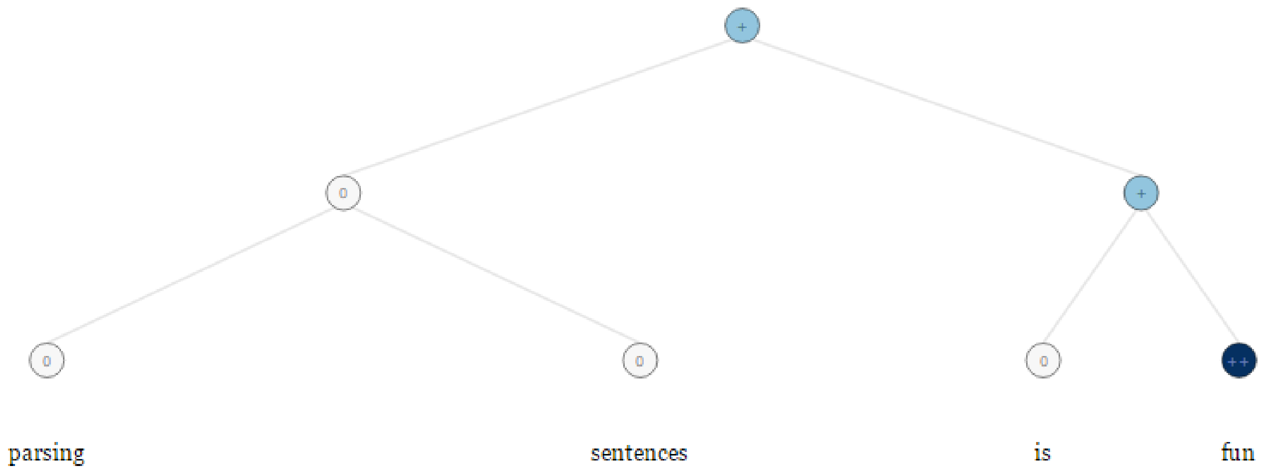
Method	**Fine-grained**	**Binary**
RAE (Socher et al., 2013)	43.2	82.4
MV-RNN (Socher et al., 2013)	44.4	82.9
RNTN (Socher et al., 2013)	45.7	85.4
DCNN (Blunsom et al., 2014)	48.5	86.8
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8
CNN-non-static (Kim, 2014)	48.0	87.2
CNN-multichannel (Kim, 2014)	47.4	**88.1**
DRNN (Irsoy and Cardie, 2014)	49.8	86.6
LSTM	46.4 (1.1)	84.9 (0.6)
Bidirectional LSTM	49.1 (1.0)	87.5 (0.5)
2-layer LSTM	46.0 (1.3)	86.3 (0.6)
2-layer Bidirectional LSTM	48.5 (1.0)	87.2 (1.0)
Dependency Tree-LSTM	48.4 (0.4)	85.7 (0.4)
Constituency Tree-LSTM		
– randomly initialized vectors	43.9 (0.6)	82.0 (0.5)
– Glove vectors, fixed	49.7 (0.4)	87.5 (0.8)
– Glove vectors, tuned	**51.0** (0.5)	88.0 (0.3)

SICK Semantic Relatedness Task

Method	Pearson's r		Spearman's ρ		MSE	
Illinois-LH (Lai and Hockenmaier, 2014)	0.7993		0.7538		0.3692	
UNAL-NLP (Jimenez et al., 2014)	0.8070		0.7489		0.3550	
Meaning Factory (Bjerva et al., 2014)	0.8268		0.7721		0.3224	
ECNU (Zhao et al., 2014)	0.8414		–		–	
Mean vectors	0.7577	(0.0013)	0.6738	(0.0027)	0.4557	(0.0090)
DT-RNN (Socher et al., 2014)	0.7923	(0.0070)	0.7319	(0.0071)	0.3822	(0.0137)
SDT-RNN (Socher et al., 2014)	0.7900	(0.0042)	0.7304	(0.0076)	0.3848	(0.0074)
LSTM	0.8528	(0.0031)	0.7911	(0.0059)	0.2831	(0.0092)
Bidirectional LSTM	0.8567	(0.0028)	0.7966	(0.0053)	0.2736	(0.0063)
2-layer LSTM	0.8515	(0.0066)	0.7896	(0.0088)	0.2838	(0.0150)
2-layer Bidirectional LSTM	0.8558	(0.0014)	0.7965	(0.0018)	0.2762	(0.0020)
Constituency Tree-LSTM	0.8582	(0.0038)	0.7966	(0.0053)	0.2734	(0.0108)
Dependency Tree-LSTM	**0.8676**	(0.0030)	**0.8083**	(0.0042)	**0.2532**	(0.0052)

Demo

- Live Demo of Sentiment Analysis
- http://nlp.stanford.edu:8080/sentiment/rntnDemo.html



Bidirectional (Lexicalized) Tree LSTM

Bidirectional Tree-Structured LSTM with Head Lexicalization

Zhiyang Teng and **Yue Zhang**
Singapore University of Technology and Design
zhiyang_teng@mymail.sutd.edu.sg
yue_zhang@sutd.edu.sg

Method	**Fine-grained**	**Binary**
RAE (Socher et al., 2013)	43.2	82.4
MV-RNN (Socher et al., 2013)	44.4	82.9
RNTN (Socher et al., 2013)	45.7	85.4
DCNN (Blunsom et al., 2014)	48.5	86.8
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8
CNN-non-static (Kim, 2014)	48.0	87.2
CNN-multichannel (Kim, 2014)	47.4	**88.1**
DRNN (Irsoy and Cardie, 2014)	49.8	86.6
LSTM	46.4 (1.1)	84.9 (0.6)
Bidirectional LSTM	49.1 (1.0)	87.5 (0.5)
2-layer LSTM	46.0 (1.3)	86.3 (0.6)
2-layer Bidirectional LSTM	48.5 (1.0)	87.2 (1.0)
Dependency Tree-LSTM	48.4 (0.4)	85.7 (0.4)
Constituency Tree-LSTM		
– randomly initialized vectors	43.9 (0.6)	82.0 (0.5)
– Glove vectors, fixed	49.7 (0.4)	87.5 (0.8)
– Glove vectors, tuned	**51.0** (0.5)	88.0 (0.3)
Bidirectional Con-Tree LSTM	**53.5**	**90.3**

Conclusions

- Can use neural ideas over parse trees
- Graph CNNs (not discussed) also exists
  - Stacking BiLSTM + Graph CNN better than both
- Is it much better?
  - remains to be seen