NLP and Deep Learning 2: Compositional Deep Learning

Christopher Manning

Stanford University

@chrmanning

2015 Deep Learning Summer School, Montreal

Compositionality

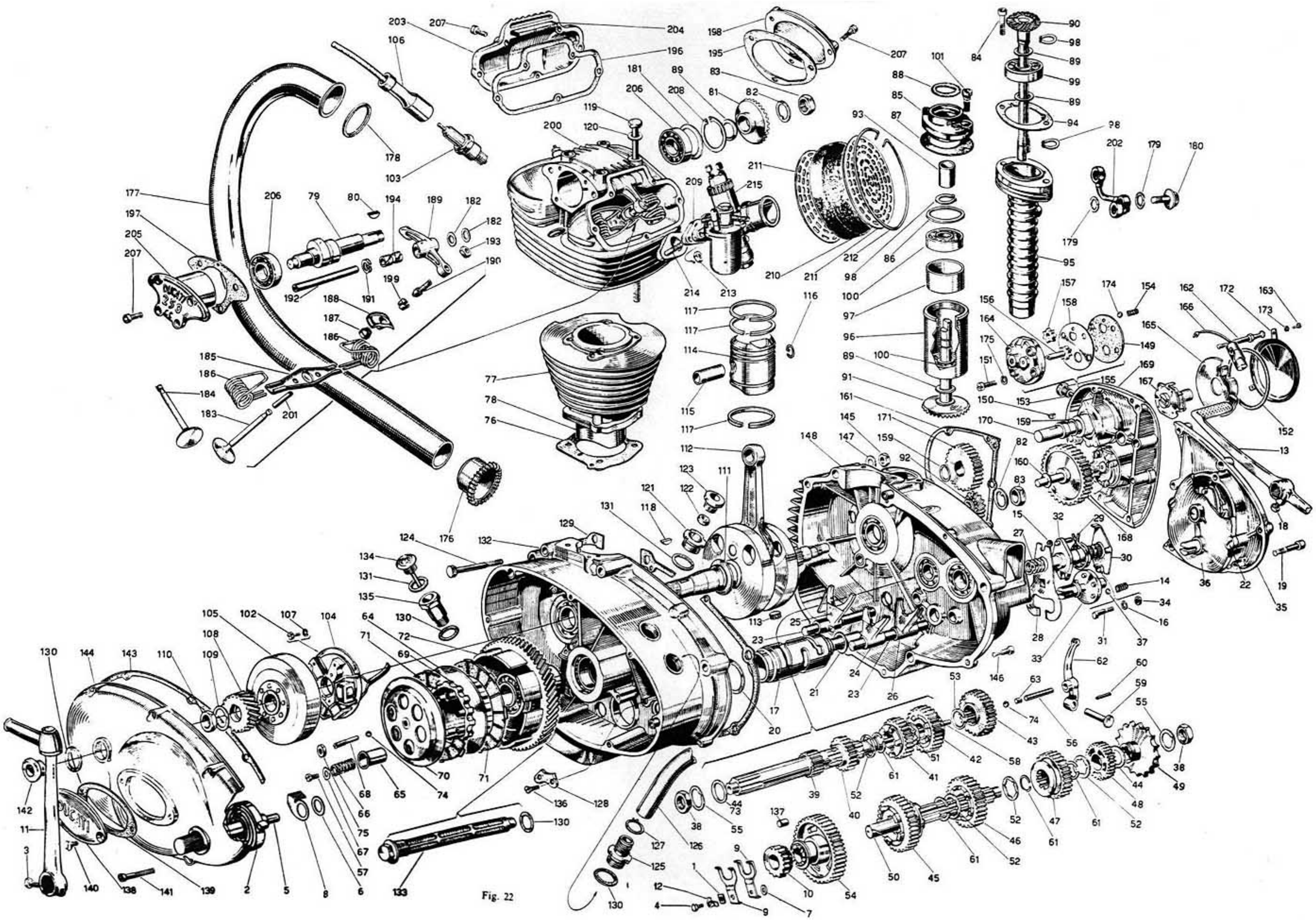


Fig. 22

Artificial Intelligence requires being able to understand bigger things from knowing about smaller parts

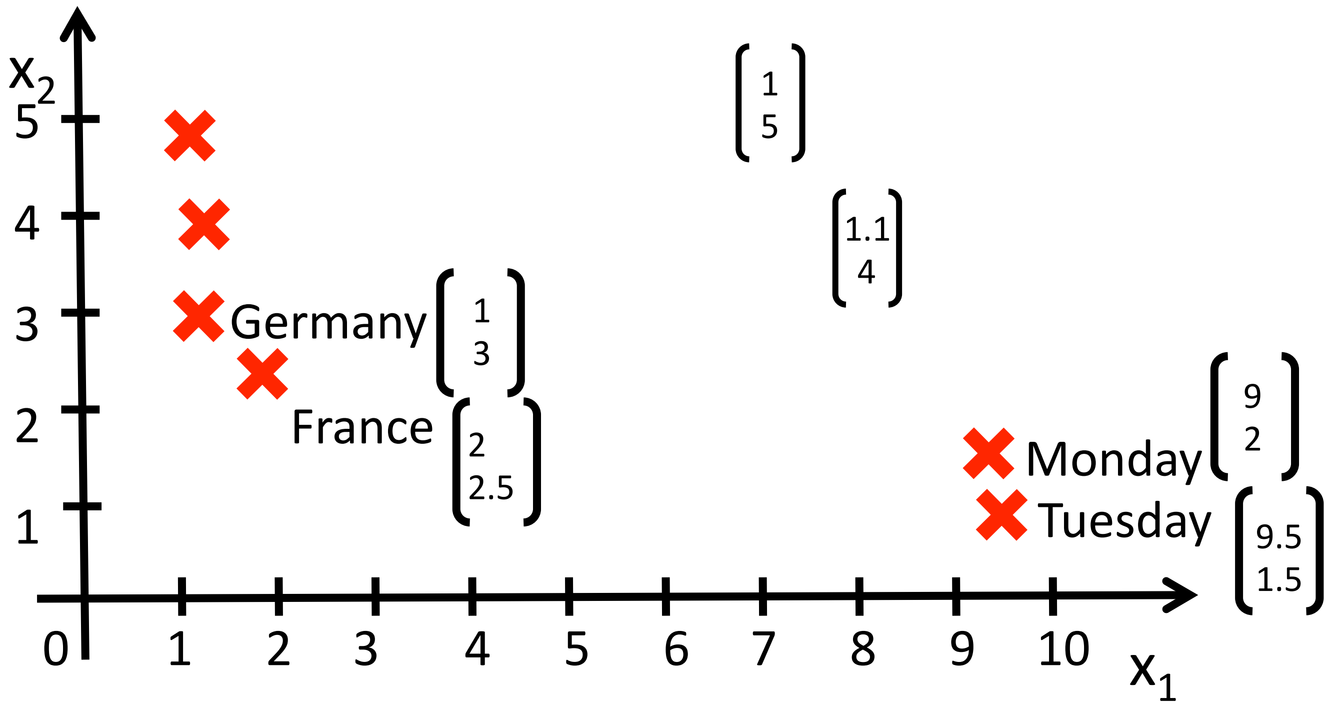
We need more than word embeddings! What of larger semantic units?

How can we know when larger units are similar in meaning?

- *The snowboarder is leaping over the mogul*
- *A person on a snowboard jumps into the air*

People interpret the meaning of larger text units – entities, descriptive terms, facts, arguments, stories – by **semantic composition** of smaller elements

Representing Phrases as Vectors



Vector for single words are useful as features but limited!

the country of my birth

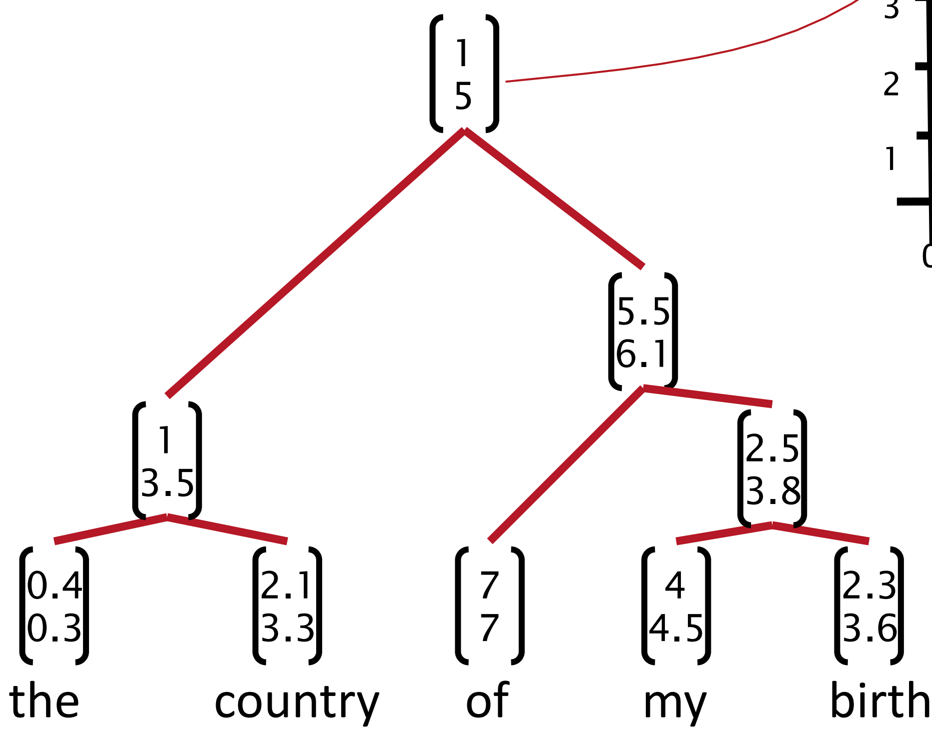
the place where I was born

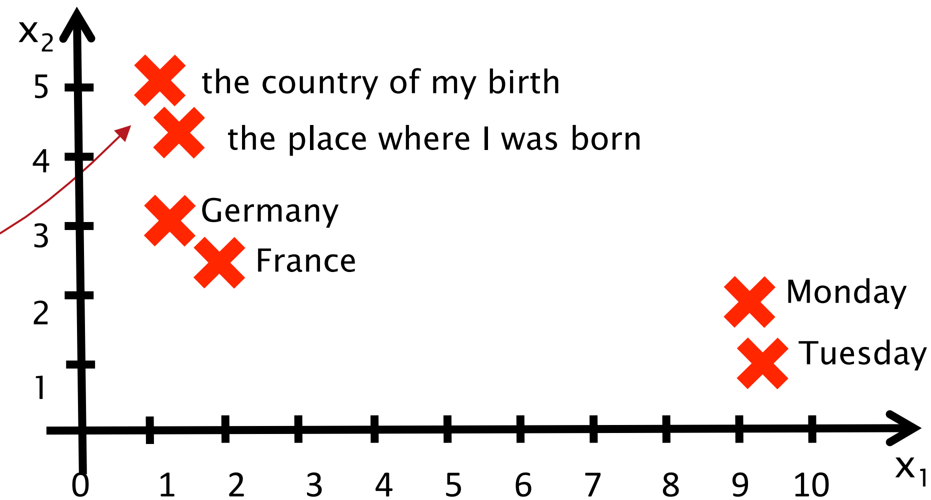
Can we extend the ideas of word vector spaces to phrases?

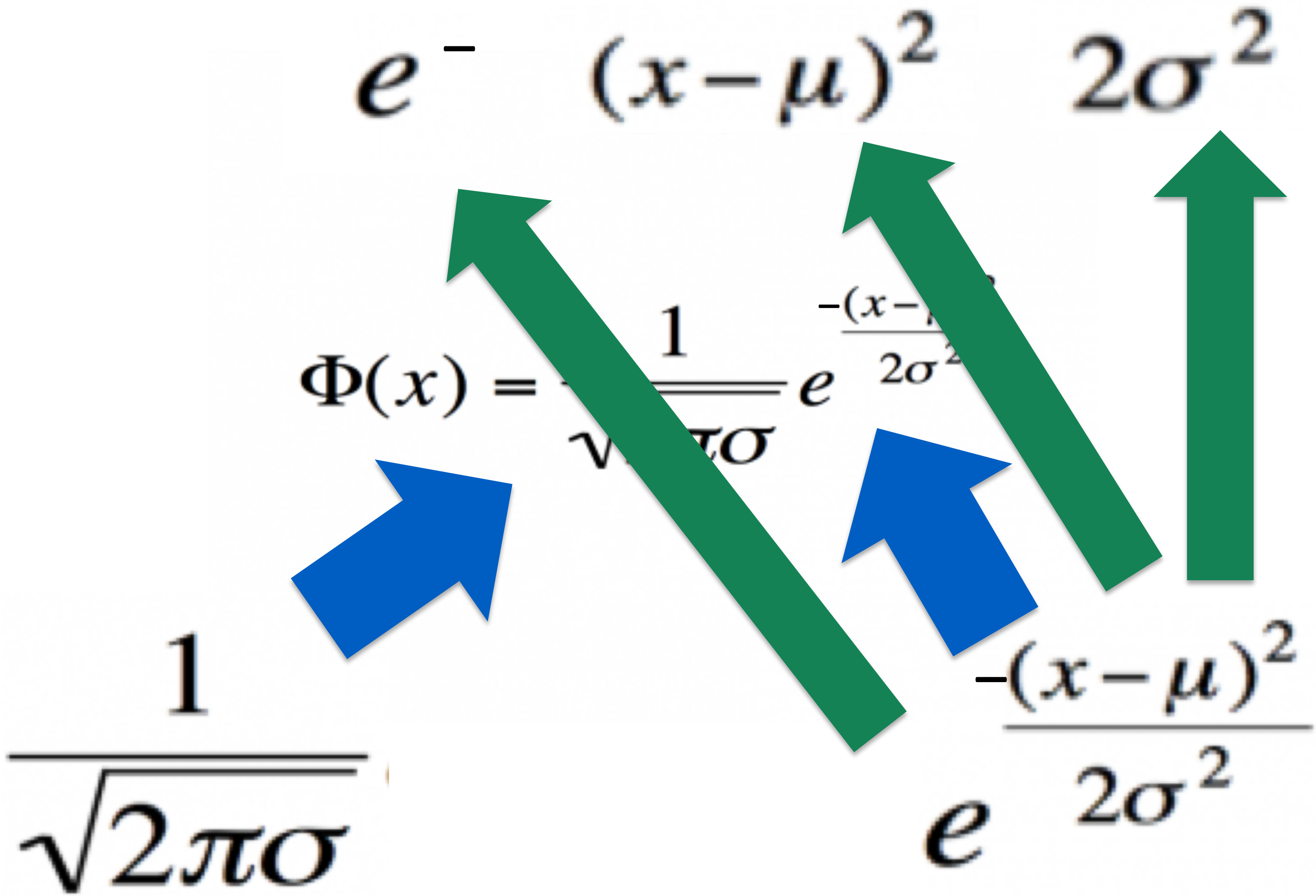
How should we map phrases into a vector space?

Use the principle of compositionality!

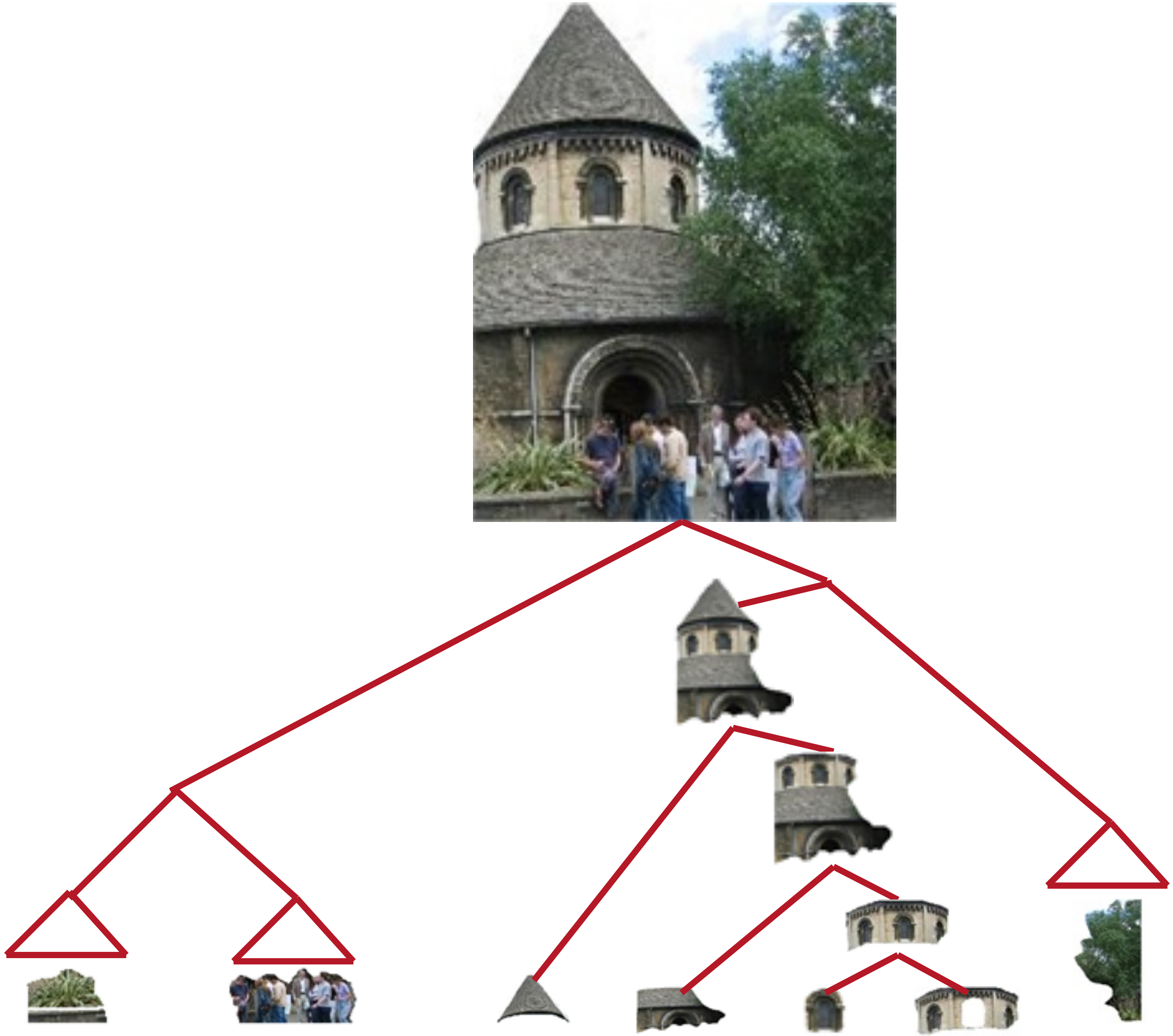
The meaning (vector) of a sentence is determined by
(1) the meanings of its words and
(2) a method that combine them.







e^{-}
$(x-\mu)^2$
$2\sigma^2$
$\Phi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
$\frac{1}{\sqrt{2\pi\sigma}}$
$e^{\frac{-(x-\mu)^2}{2\sigma^2}}$



Can we build
meaning composition functions
in deep learning systems?

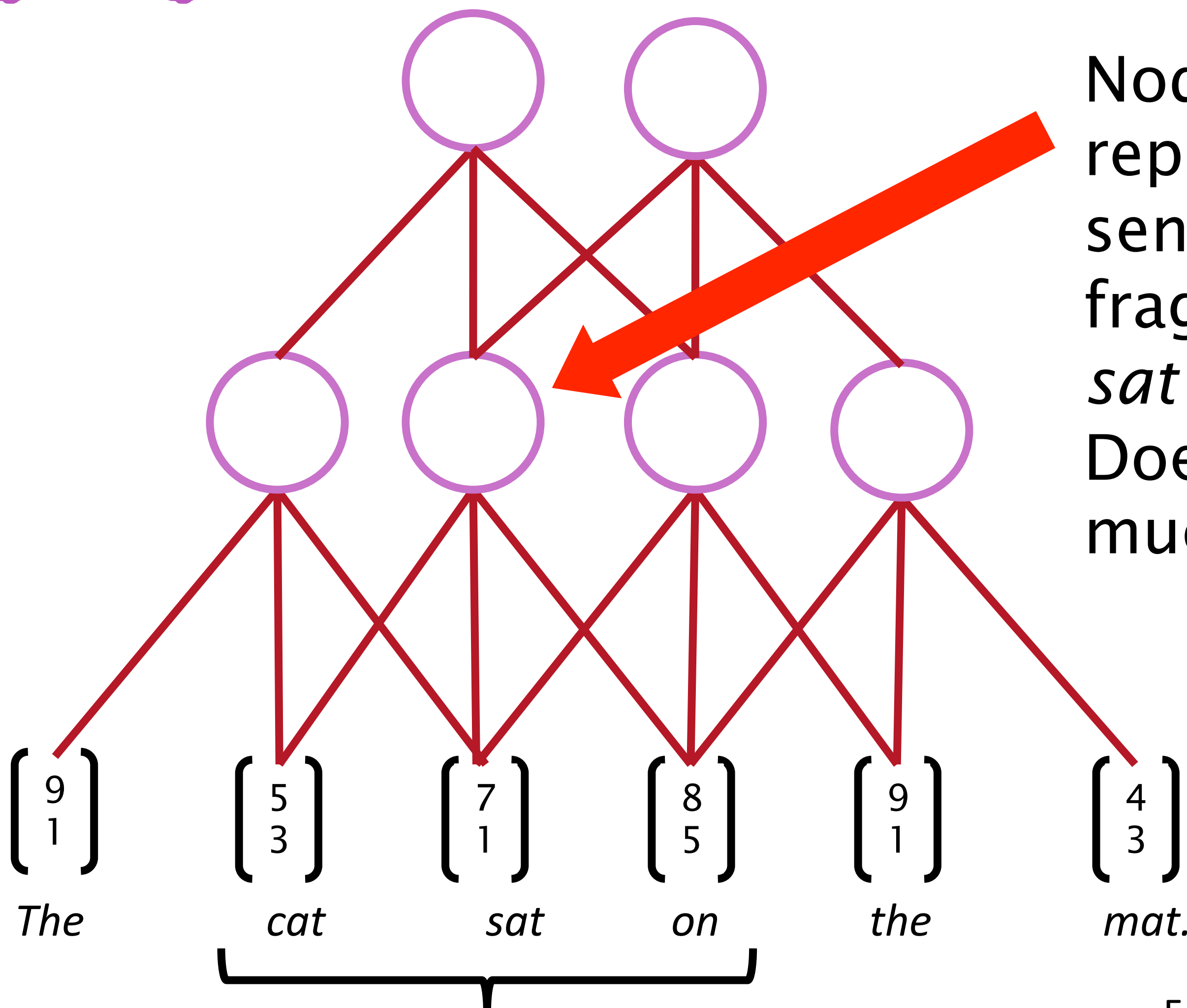
Conjecture

You can attempt to model language with a simple, uniform architecture

- A sequence model (RNN, LSTM, …)
- A 1d convolutional neural network

However, maybe one can produce a better composition function for language by modeling an input-specific compositional tree

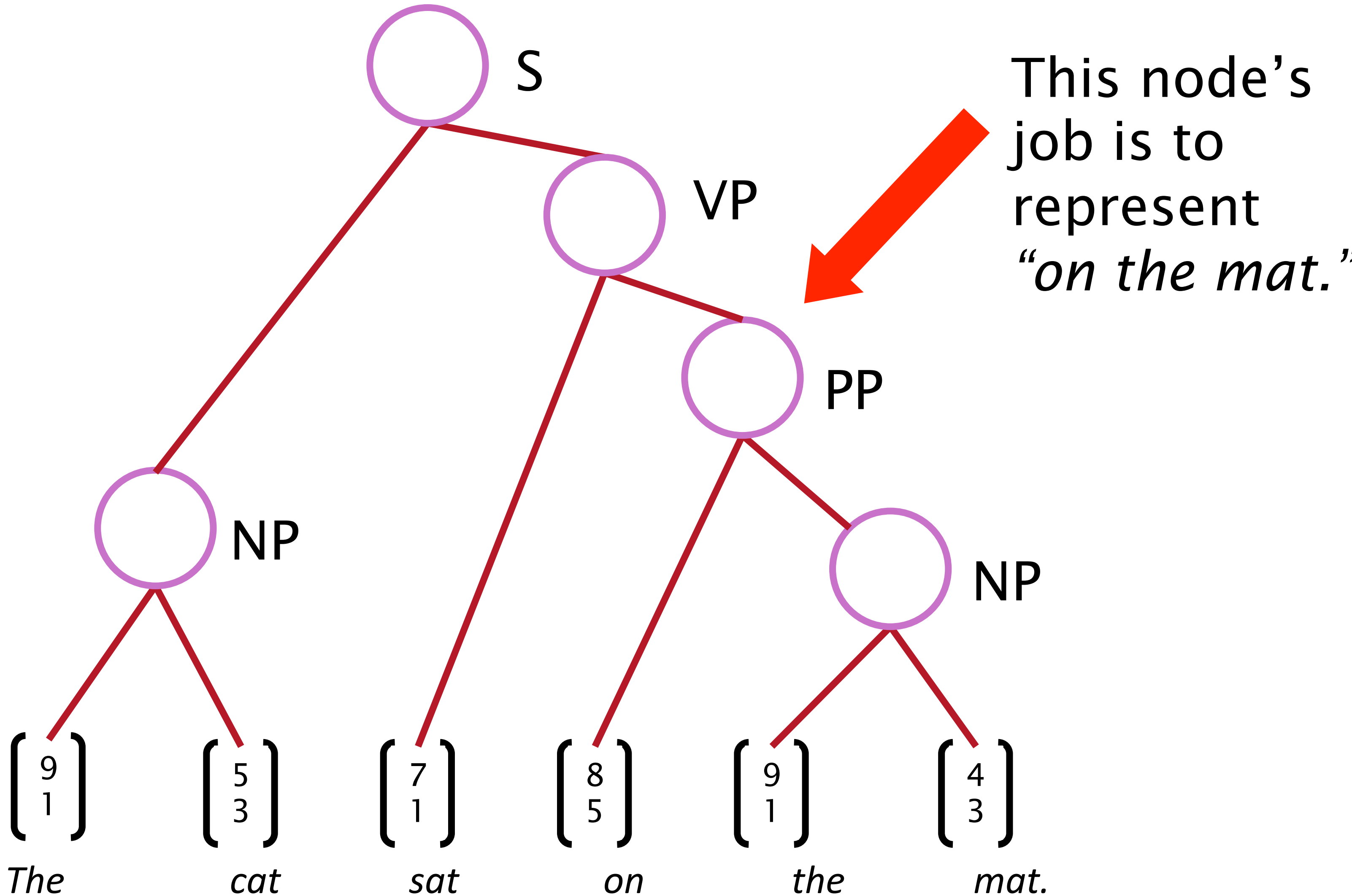
A "generic" hierarchy on natural language doesn't make sense?



Node has to represent sentence fragment "*cat sat on.*" Doesn't make much sense.

Feature representation for words

What we want:
An input-dependent tree structure



Strong priors? Universals of language?

- This is a controversial issue (read: **Chomsky!**), but there does seem to be a fairly common structure over all human languages
  - Much of this may be functionally motivated
- To what extent should we use these priors in our ML models?

Universal 18 [Greenberg 63]

N, Adj word order in the world's languages

	N-Adj	Adj-N
Num-N	17%	**27%**
N-Num	**52%**	*4%

"Harmonic" patterns (which preserve order of N) are **favored**

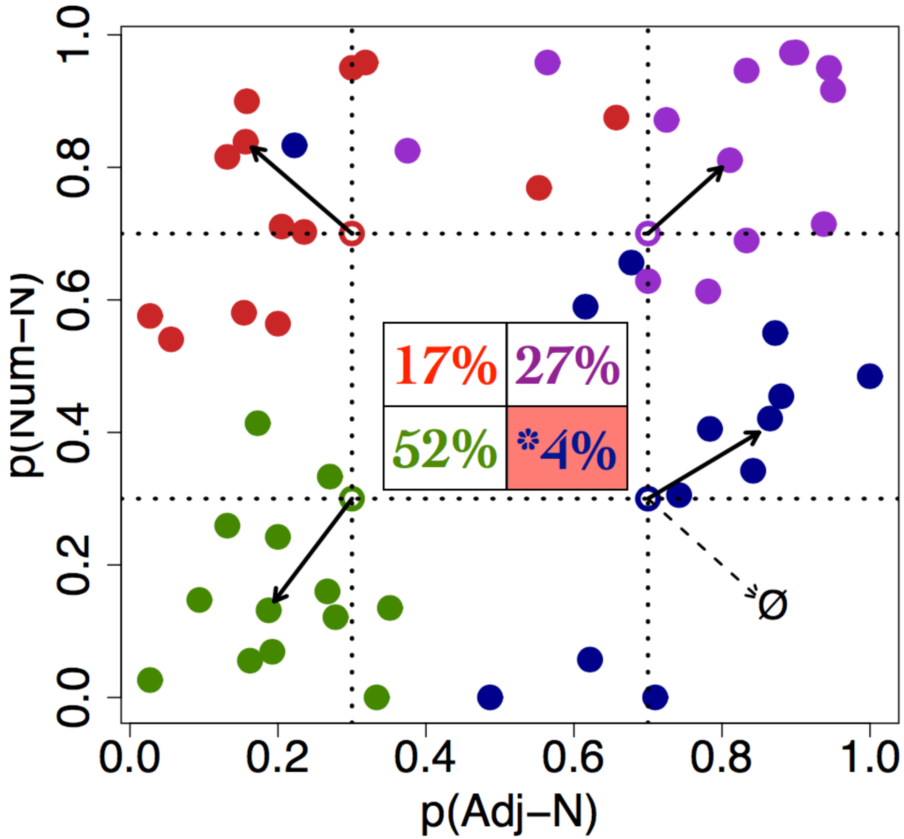
*Particular non-harmonic pattern Adj-N, N-Num is **disfavored**

Substantive learning bias

Dryer, M. (2008a). Order of adjective and noun. In M. Haspelmath, M. S. Dryer, D. Gil, & B. Comrie (Eds.), *The World Atlas of Language Structures Online*, Chapter 87. Munich, Germany: Max Planck Digital Library.

Experimental results I: Adult individual data

Do typological statistics correspond to any active on-line learning bias?



Learners of an artificial language, given two-word nonce-utterance examples (N with either Adj or Num), dominant order in each of 4 conditions = 70%.

Culbertson, J., Smolensky, P. & Wilson, C. 2013. Cognitive biases, linguistic universals, and constraint-based grammar learning. *Topics in Cognitive Science*, 5, 392–424.

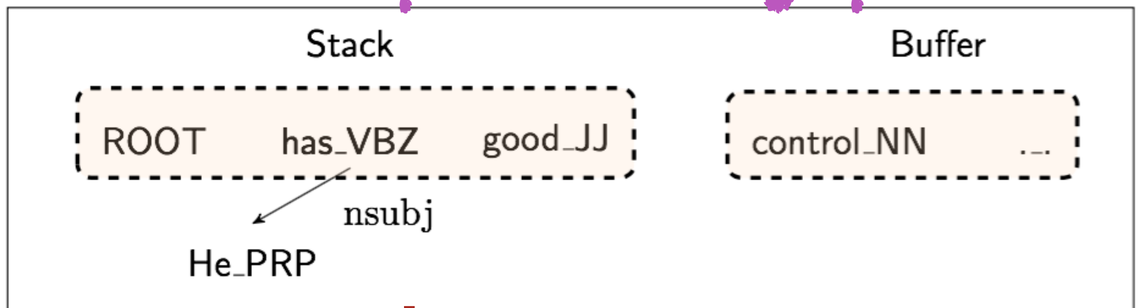
Where does the tree structure come from?

1. It can come from a conventional statistical NLP parser, such as the Stanford Parser's PCFG
2. It can be built by a neural network component, such as a neural network dependency parser [advertisement]
3. It can be learned and built as part of the training/operation of the TreeRNN system, by adding another matrix to score the goodness of constituents built

Mainly, we've done 1 or 2.

Transition-based dependency parsers

Decide next move from configuration



Indicator features binary, sparse dim $=10^6 \sim 10^7$

0	0	0	1	0	0	1	0	...	0	0	1	0

Feature templates: usually a combination of 1 ~ 3 elements from the configuration.

$$s1.w = \text{good} \wedge s1.t = \text{JJ}$$

$$s2.w = \text{has} \wedge s2.t = \text{VBZ} \wedge s1.w = \text{good}$$

$$lc(s_2).t = \text{PRP} \wedge s_2.t = \text{VBZ} \wedge s_1.t = \text{JJ}$$

$$lc(s_2).w = \text{He} \wedge lc(s_2).l = \text{nsubj} \wedge s_2.w = \text{has}$$

Sparse!

Incomplete!

Slow!!! (95% of time)

Deep Learning Dependency Parser

[Chen & Manning, EMNLP 2014]

http://nlp.stanford.edu/software/nndep.shtml

Softmax layer:

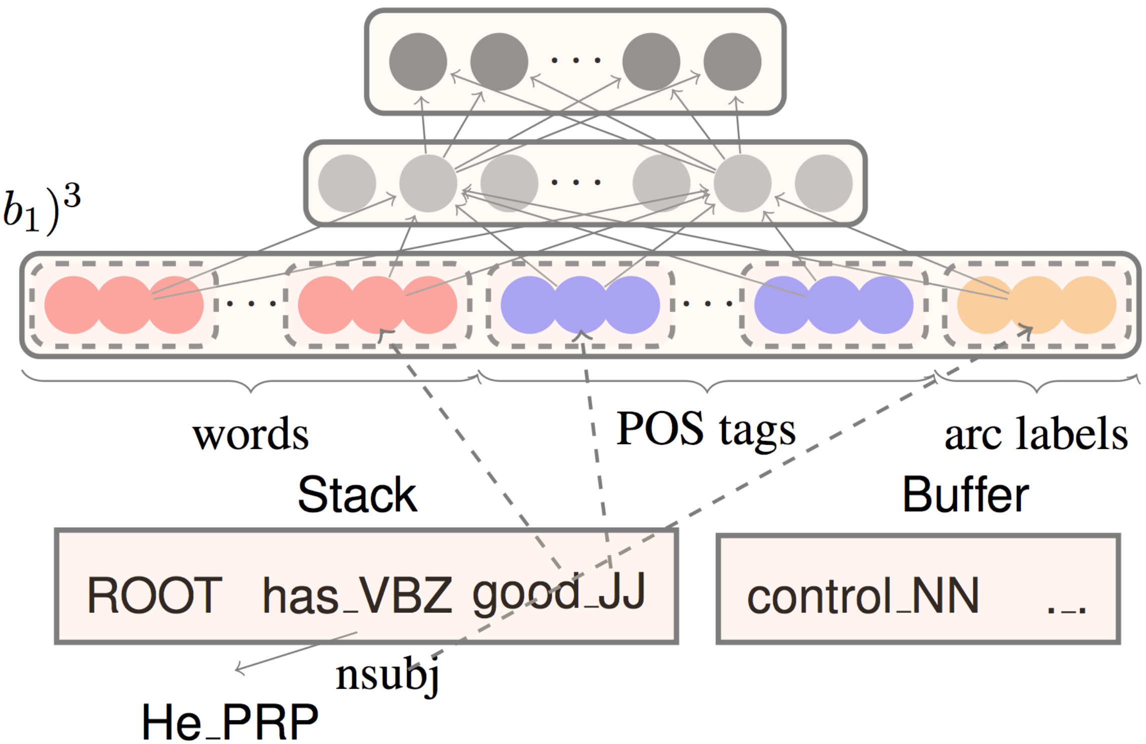
$$p = \mathtt{softmax}(W_2 h)$$

Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$

Configuration



Deep Learning Dependency Parser

[Chen & Manning, EMNLP 2014]

- An accurate and fast neural-network-based dependency parser!
- Parsing to Stanford Dependencies:
  - Unlabeled attachment score (UAS) = head
  - Labeled attachment score (LAS) = head and label

Parser	UAS	LAS	sent / s
MaltParser	89.8	87.2	469

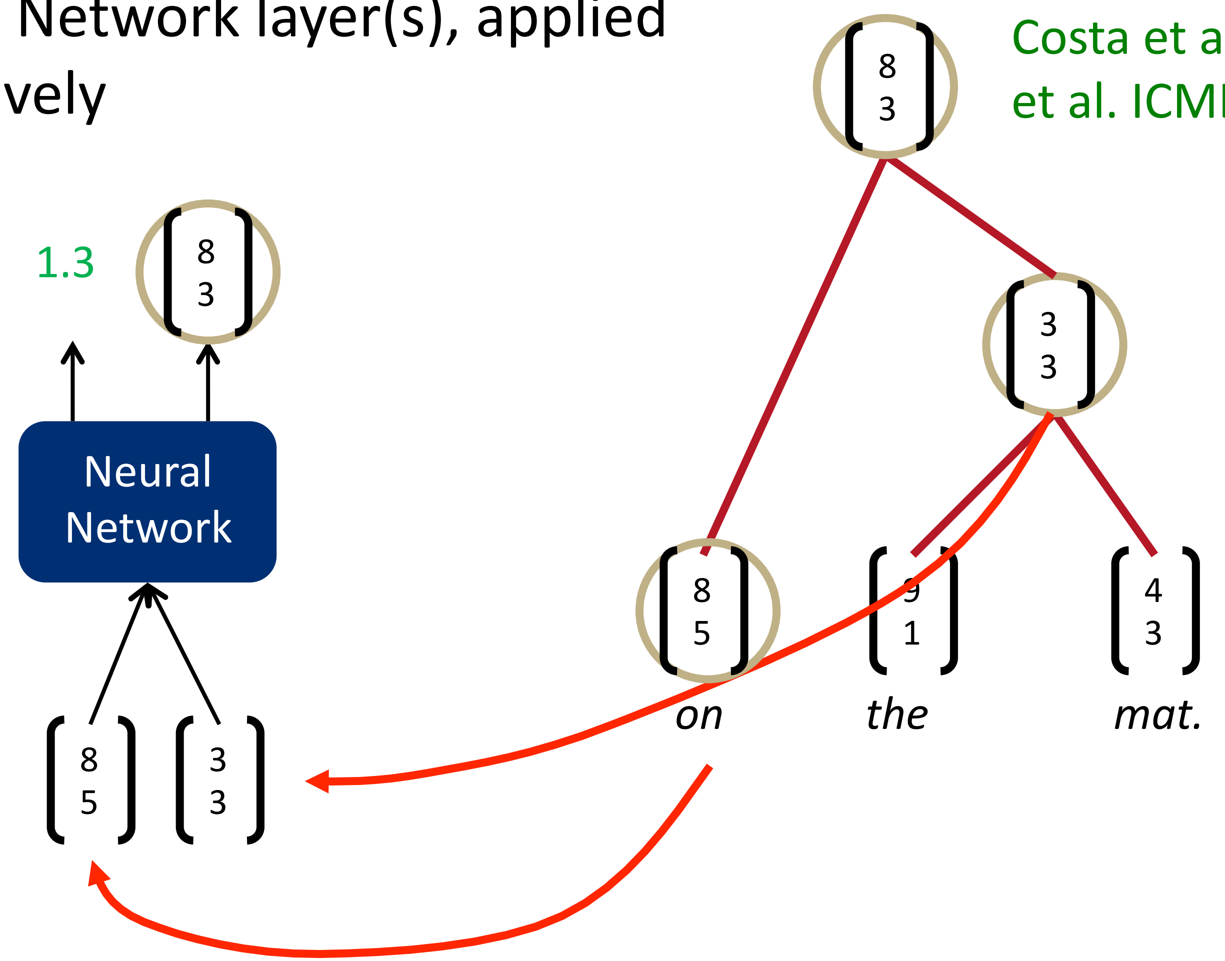
Google pulling out all the stops 94.3 92.4

Five attempts at meaning composition

Tree Recursive Neural Networks (Tree RNNs)

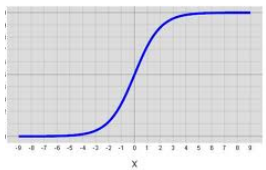
Computational unit:
Neural Network layer(s), applied recursively

(Goller & Küchler 1996, Costa et al. 2003, Socher et al. ICML, 2011)

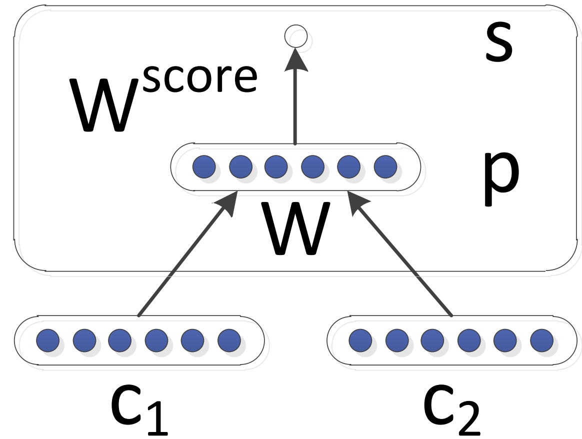


Version 1: Simple concatenation Tree RNN

$$p = \tanh\left(W\begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),$$

where tanh: 

$$\text{score} = W^{score} p$$



Earlier TreeRNN work includes (Goller & Küchler 1996), with a fixed tree structure, Costa et al. (2003) using an RNN for PP attachment, but on one hot vectors, Bottou (2011) for compositionality with recursion

Semantic similarity: nearest neighbors

All the figures are adjusted for seasonal variations
1. All the numbers are adjusted for seasonal fluctuations
2. All the figures are adjusted to remove usual seasonal patterns

Knight-Ridder would n't comment on the offer
1. Harsco declined to say what country placed the order
2. Coastal would n't disclose the terms

Sales grew almost 7% to $UNK m. from $UNK m.
1. Sales rose more than 7% to $94.9 m. from $88.3 m.
2. Sales surged 40% to UNK b. yen from UNK b.

Version 1 Limitations

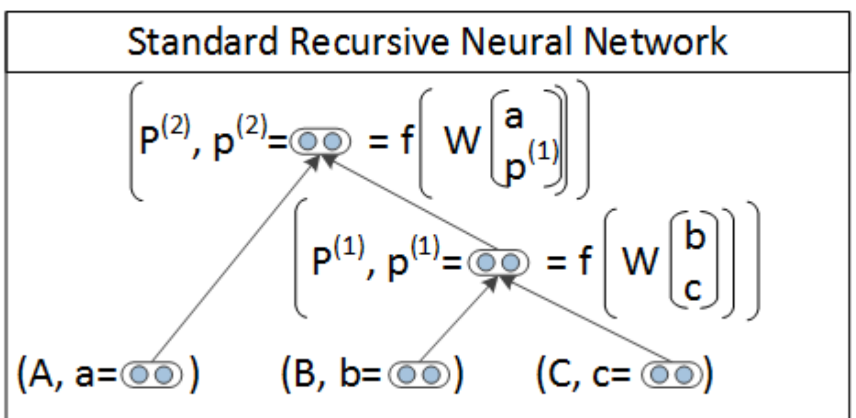
Composition function is a single weight matrix!

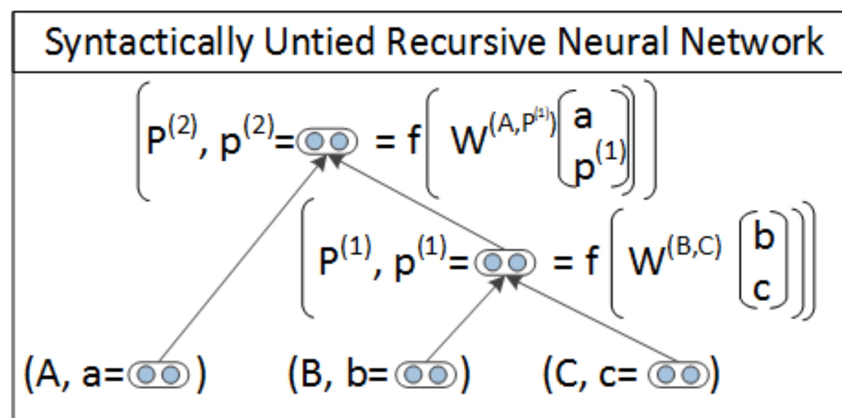
No real interaction between the input words!

Not adequate for human language composition function

Version 2: PCFG + Syntactically-Untied RNN

- A symbolic Context-Free Grammar (CFG) backbone is adequate for basic syntactic structure
- We use the discrete syntactic categories of the children to choose the composition matrix
- An RNN can do better with a different composition matrix for different syntactic environments
- The result gives us a better semantics





Experiments

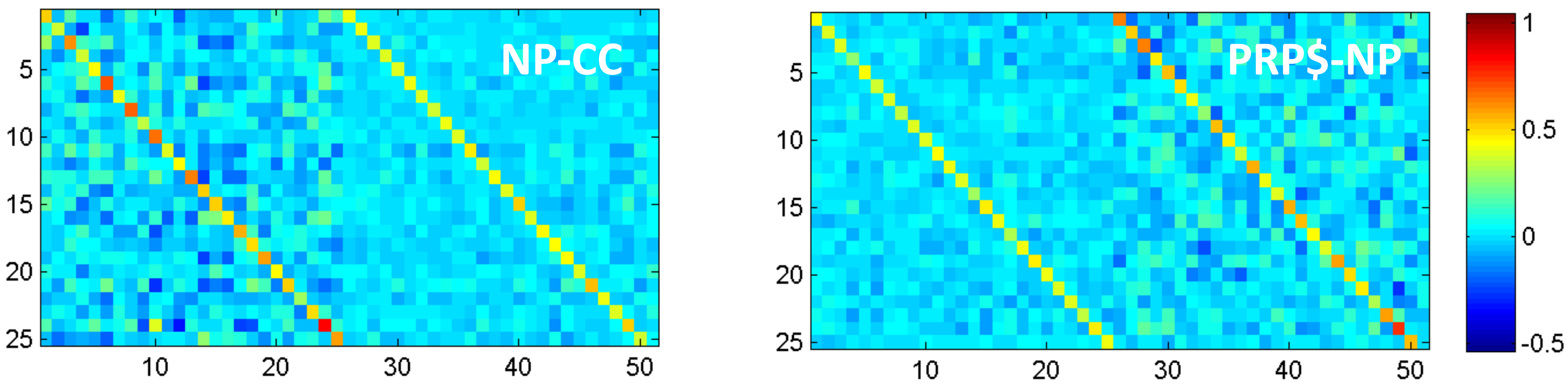
Parser	Test, All Sentences
Stanford PCFG, (Klein and Manning, 2003a)	85.5
Stanford Factored (Klein and Manning, 2003b)	86.6
Factored PCFGs (Hall and Klein, 2012)	89.4
Collins (Collins, 1997)	87.7
SSN (Henderson, 2004)	89.4
Berkeley Parser (Petrov and Klein, 2007)	90.1
CVG (RNN) (Socher et al., ACL 2013)	85.0
CVG (SU-RNN) (Socher et al., ACL 2013)	90.4
Charniak - Self Trained (McClosky et al. 2006)	91.0
Charniak - Self Trained-ReRanked (McClosky et al. 2006)	92.1

Standard *WSJ* split, labeled F_1

SU-RNN / CVG
[Socher, Bauer, Manning, Ng 2013]

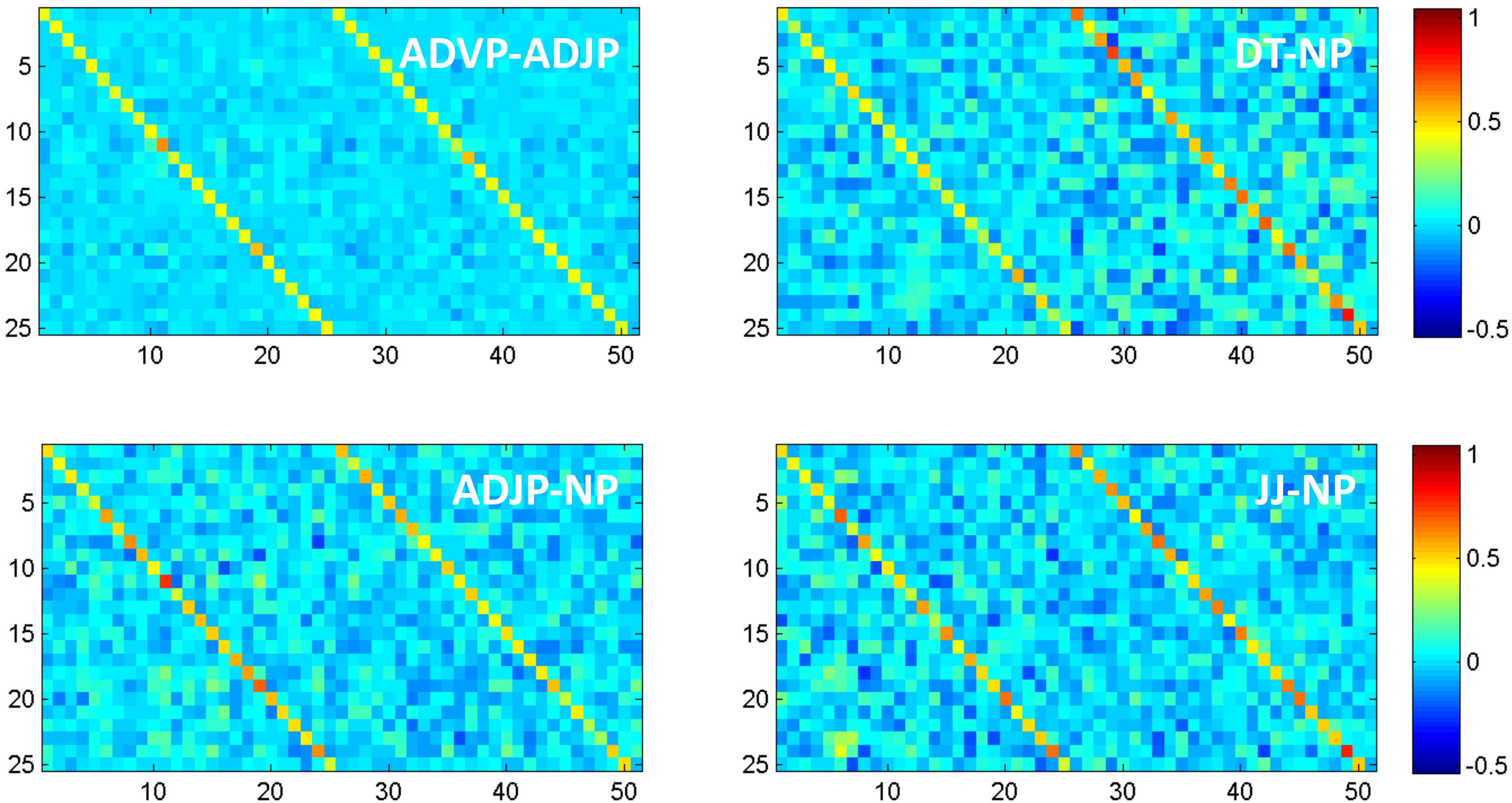
Learns soft notion of head words

Initialization: $W^{(\cdot\cdot)} = 0.5[I_{n\times n} I_{n\times n} 0_{n\times 1}] + \epsilon$



SU-RNN / CVG

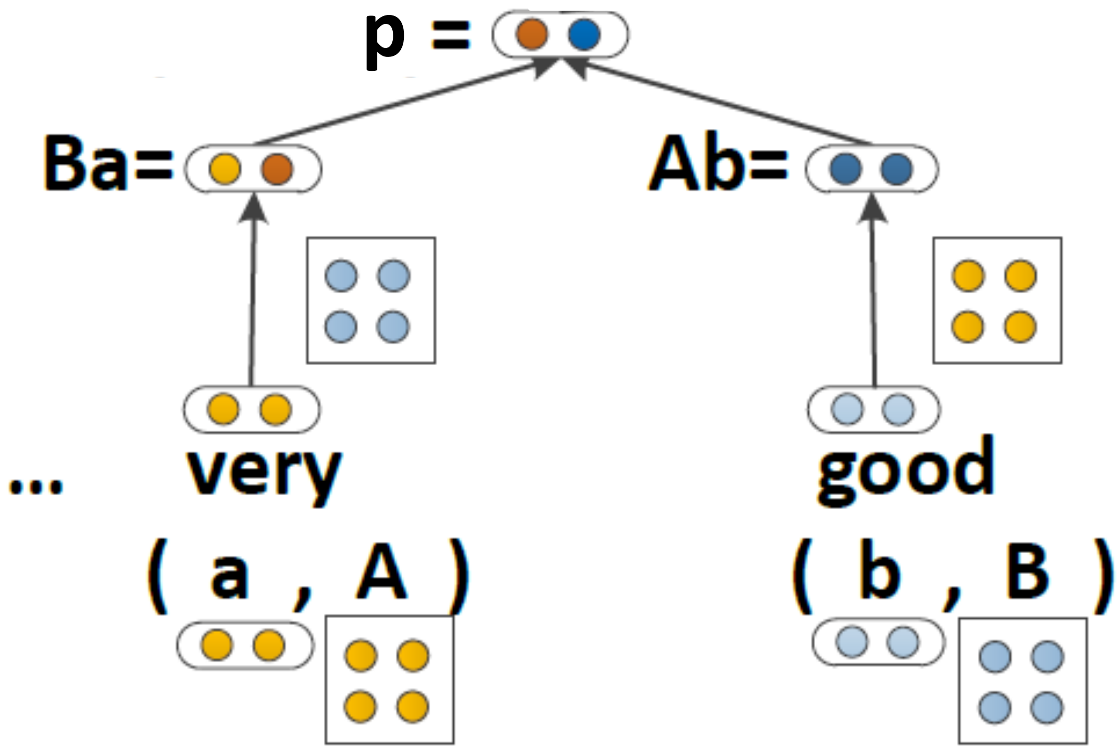
[Socher, Bauer, Manning, Ng 2013]



Version 3: Matrix-vector RNNs
[Socher, Huval, Bhat, Manning, & Ng, 2012]

$$p = f\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

$$p = f\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$



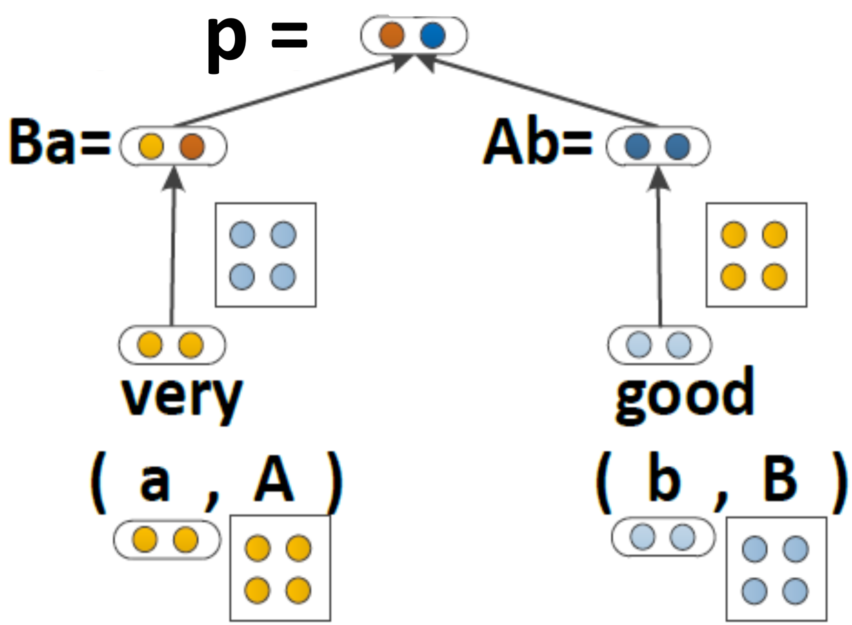
Version 3: Matrix-vector RNNs

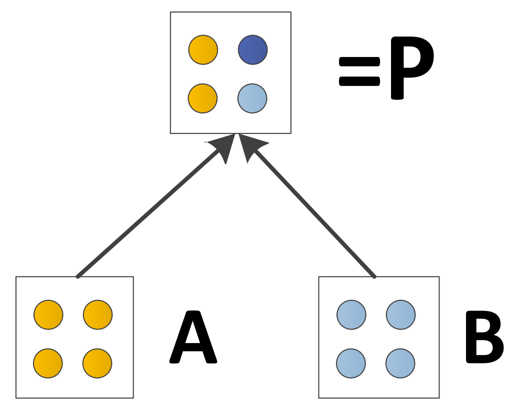
[Socher, Huval, Bhat, Manning, & Ng, 2012]

$$p = f\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}\right)$$

$$P = g(A, B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$$

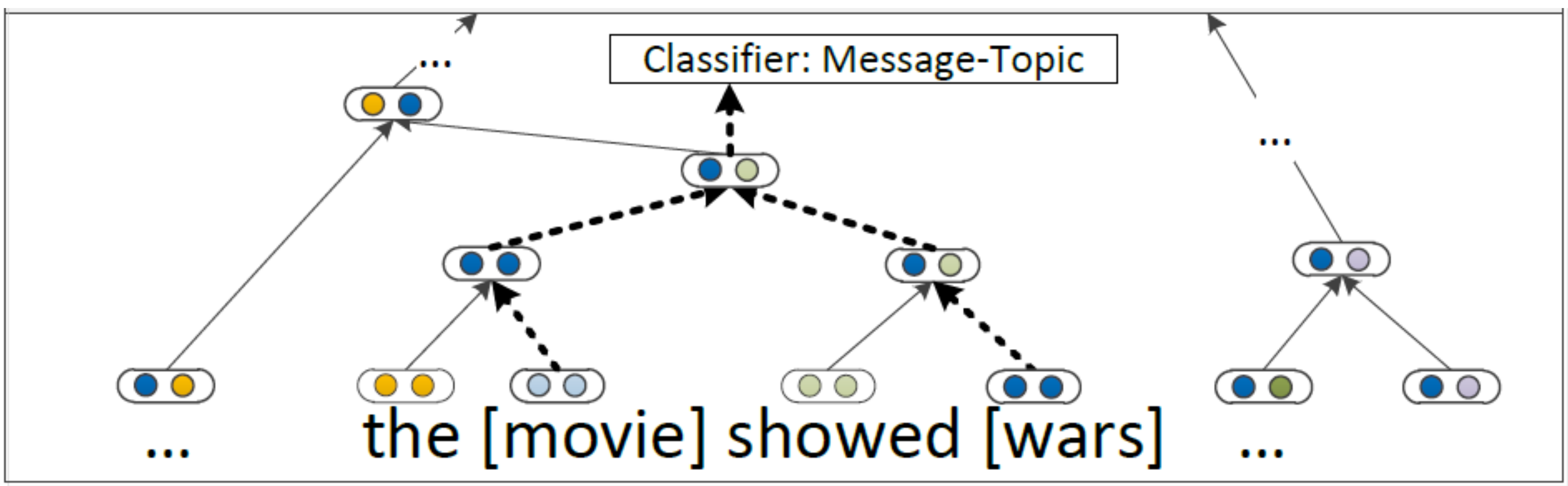
$$W_M \in \mathbb{R}^{n \times 2n}$$





Classification of Semantic Relationships

- Can an MV-RNN learn how a large syntactic context conveys a semantic relationship?
- My [apartment]$_{e1}$ has a pretty large [kitchen]$_{e2}$ → component-whole relationship (e2,e1)
- Build a single compositional semantics for the minimal constituent including both terms

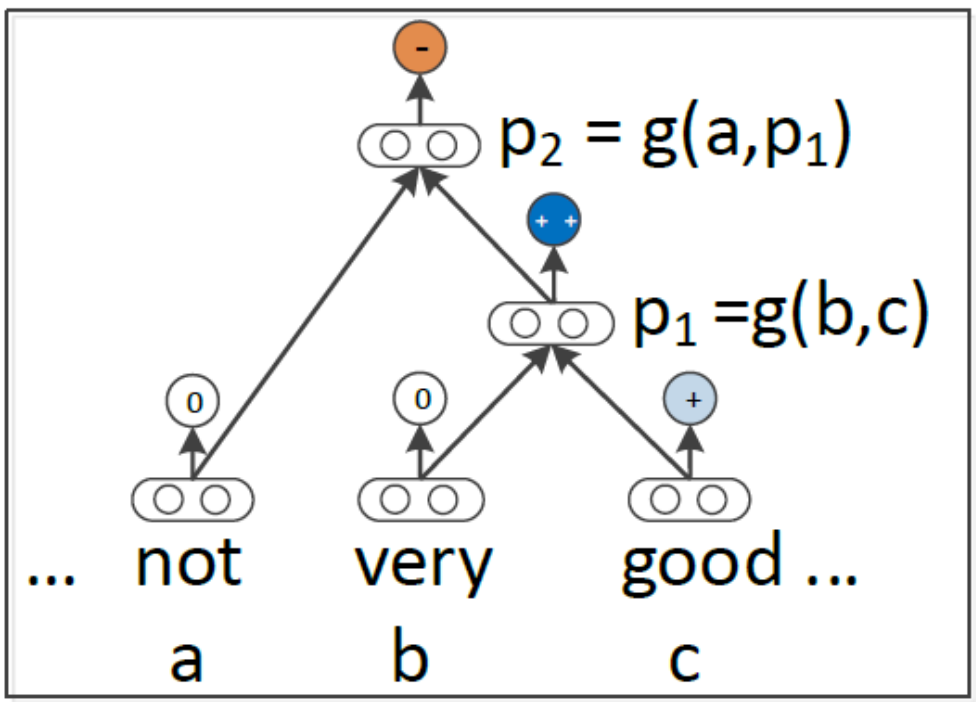


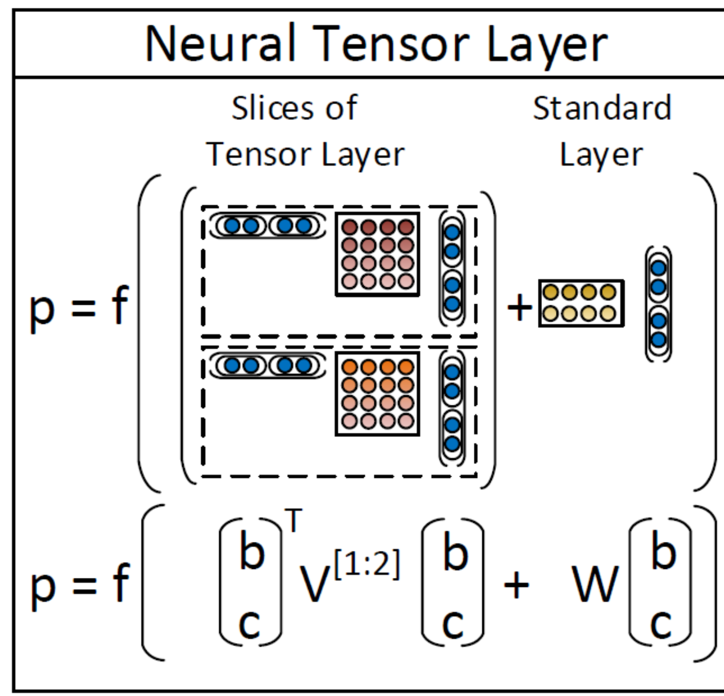
Classification of Semantic Relationships

Classifier	Features	F1
SVM	POS, stemming, syntactic patterns	60.1
MaxEnt	POS, WordNet, morphological features, noun compound system, thesauri, Google n-grams	77.6
SVM	POS, WordNet, prefixes, morphological features, dependency parse features, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner	82.2
RNN	–	74.8
MV-RNN	–	79.1
MV-RNN	**POS, WordNet, NER**	**82.4**

Version 4: Recursive Neural Tensor Network

- Less parameters than MV-RNN
- Allows the two word or phrase vectors to interact multiplicatively





Beyond the bag of words: Sentiment detection

Is the tone of a piece of text positive, negative, or neutral?

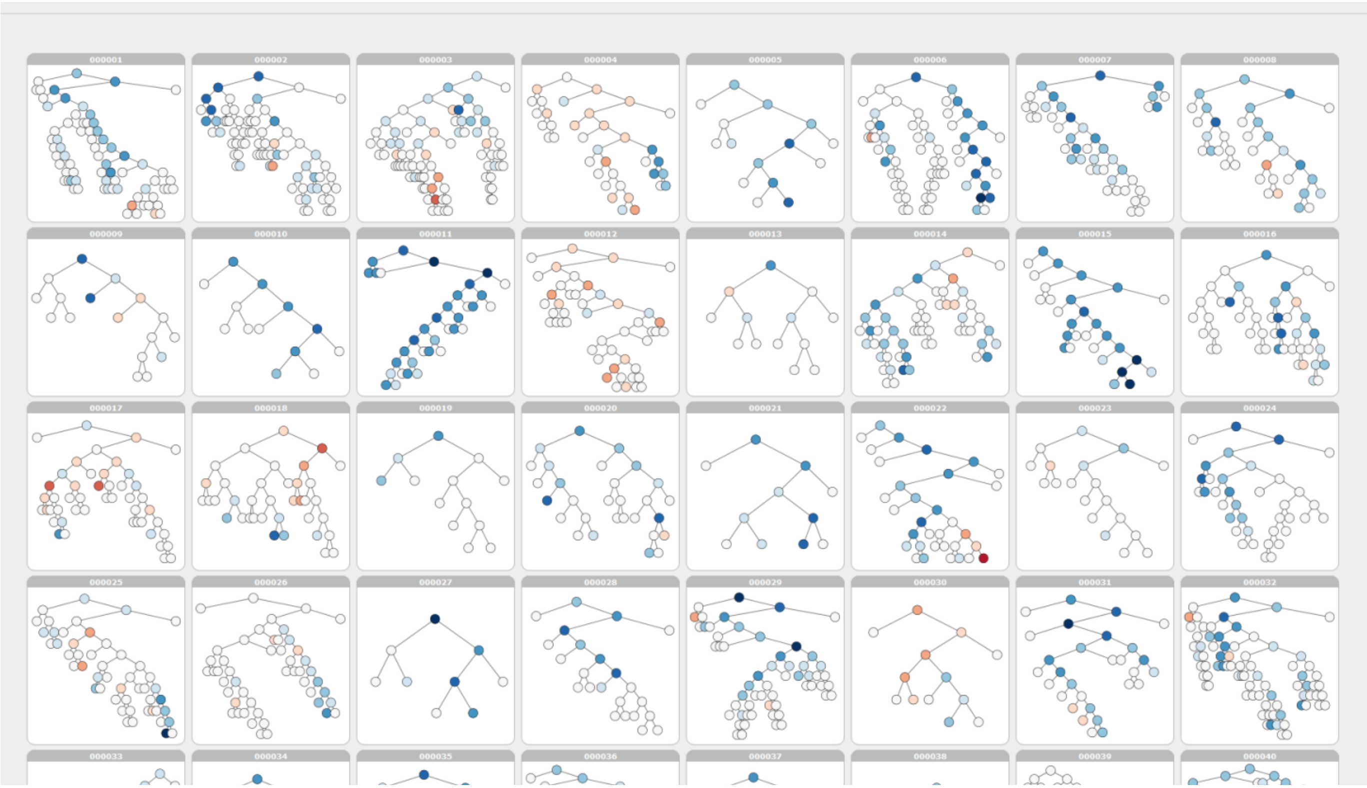
- Sentiment is that sentiment is “easy”
- Detection accuracy for longer documents ~90%, BUT

… … loved … … … … … great … … … … … … impressed … … … … … … marvelous … … … …

With this cast, and this subject matter, the movie should have been funnier and more entertaining.

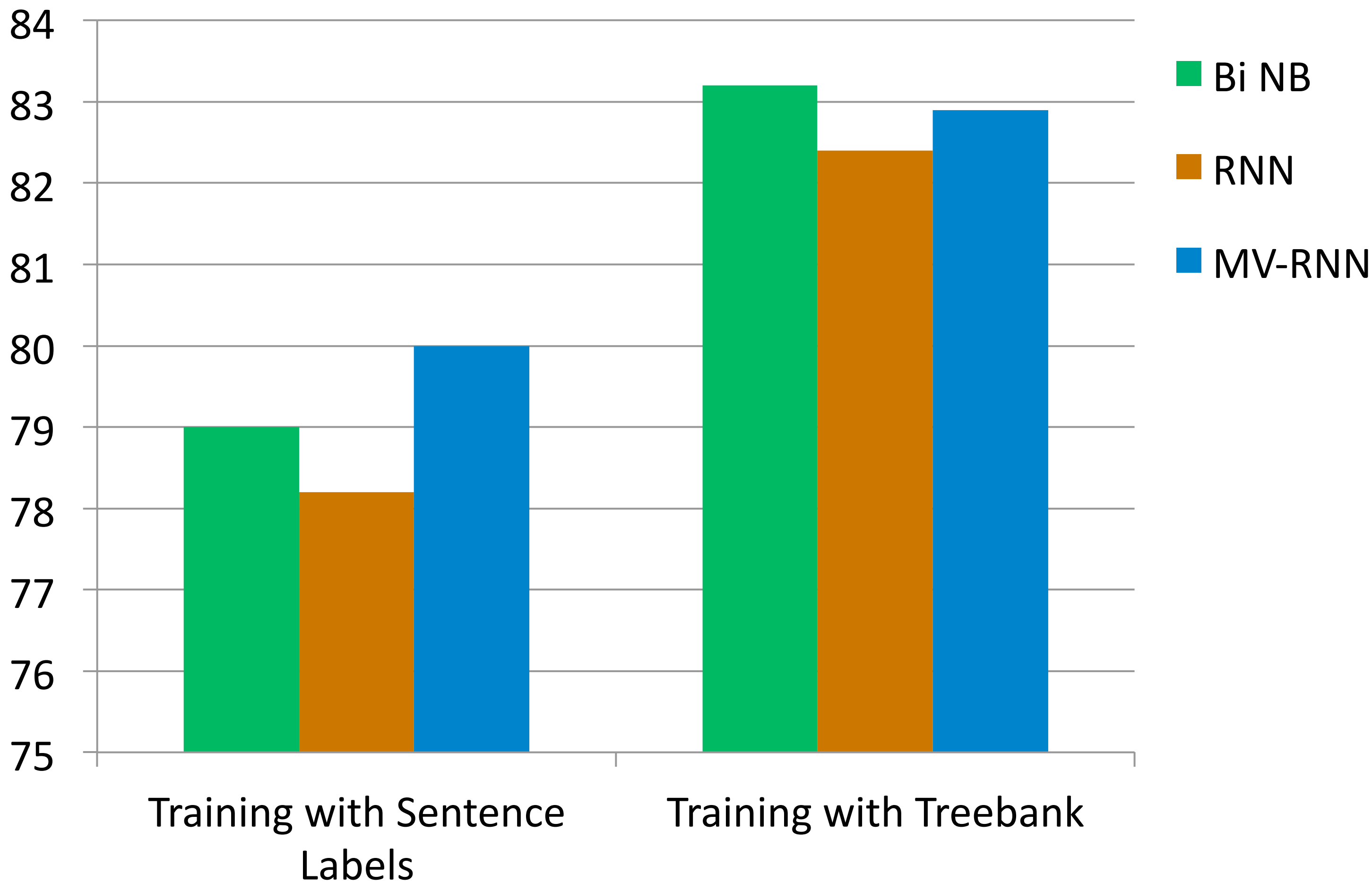
Stanford Sentiment Treebank

- 215,154 phrases labeled in 11,855 sentences
- Can actually train and test compositions



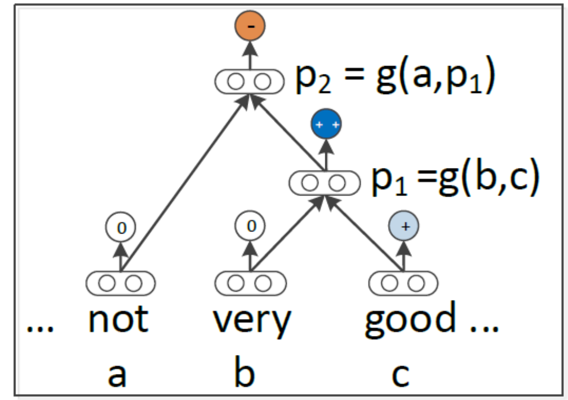
http://nlp.stanford.edu:8080/sentiment/

Better Dataset Helped All Models

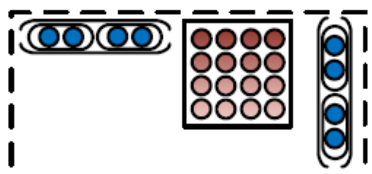


- Hard negation cases are still mostly incorrect
- We also need a more powerful model!

Version 4: Recursive Neural Tensor Network

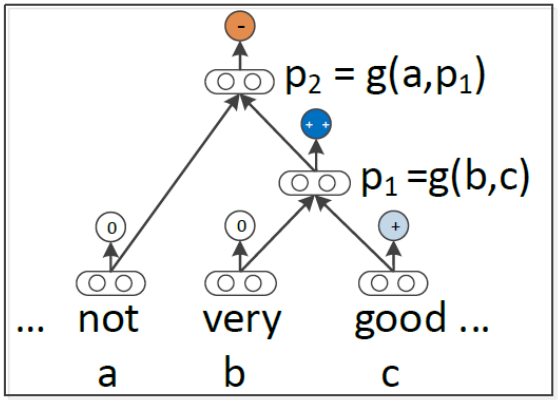


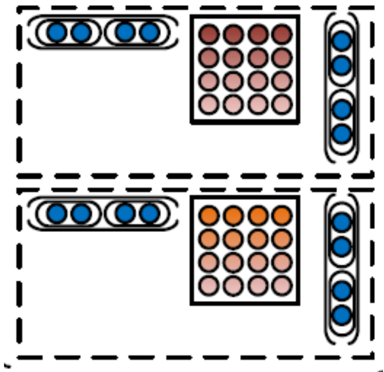
Idea: Allow both additive and mediated multiplicative interactions of vectors



$$\begin{bmatrix} b \\ c \end{bmatrix}^T V \begin{bmatrix} b \\ c \end{bmatrix}$$

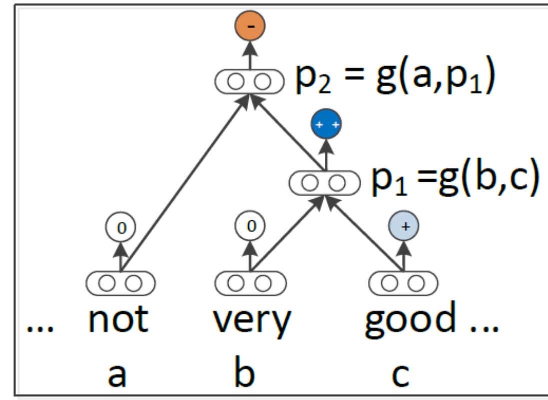
Recursive Neural Tensor Network

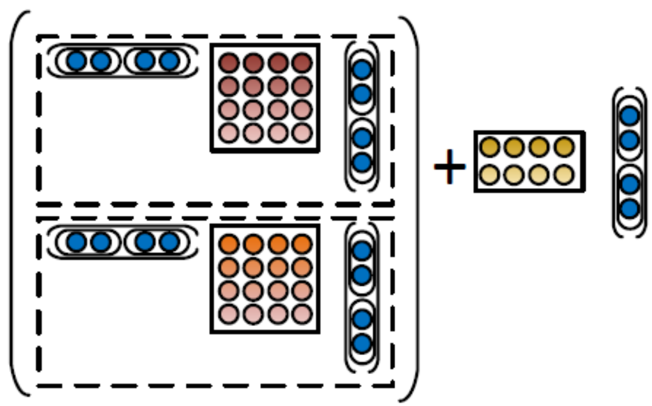




$$\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:2]} \begin{bmatrix} b \\ c \end{bmatrix}$$

Recursive Neural Tensor Network

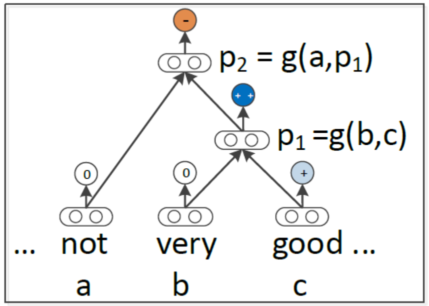


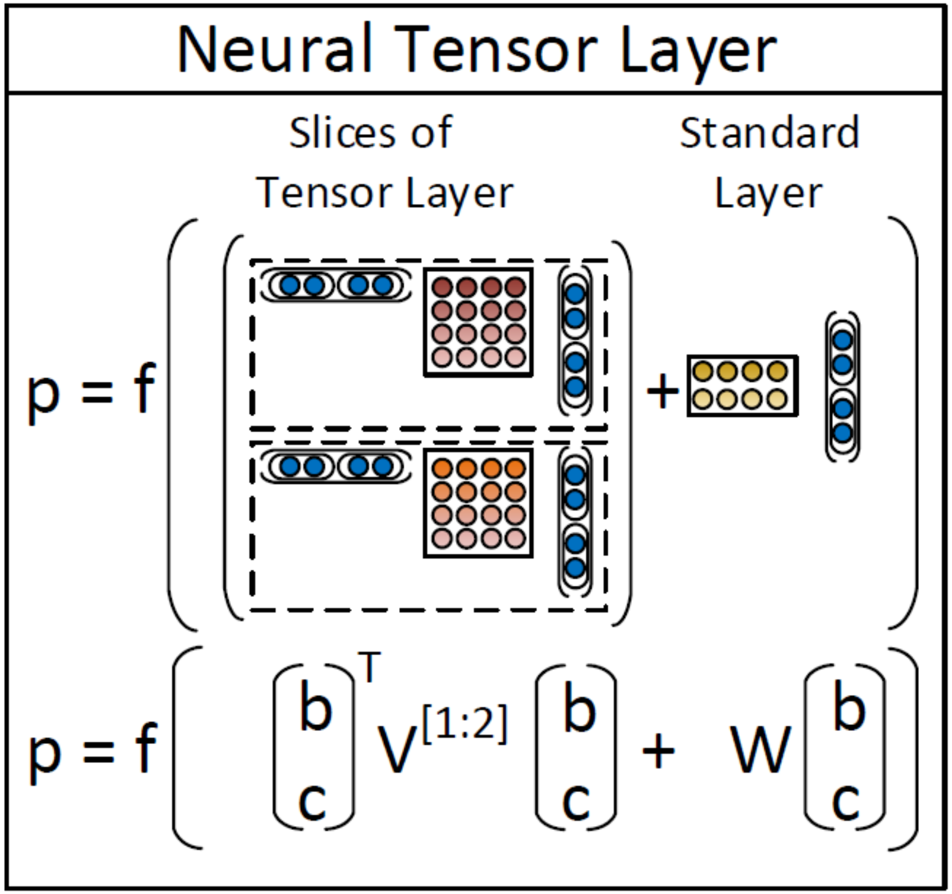


$$\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:2]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix}$$

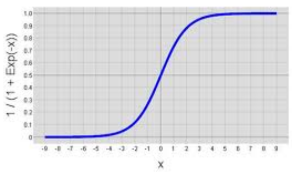
Recursive Neural Tensor Network

- Use resulting vectors in tree as input to a classifier like logistic regression
- Train all weights jointly with gradient descent



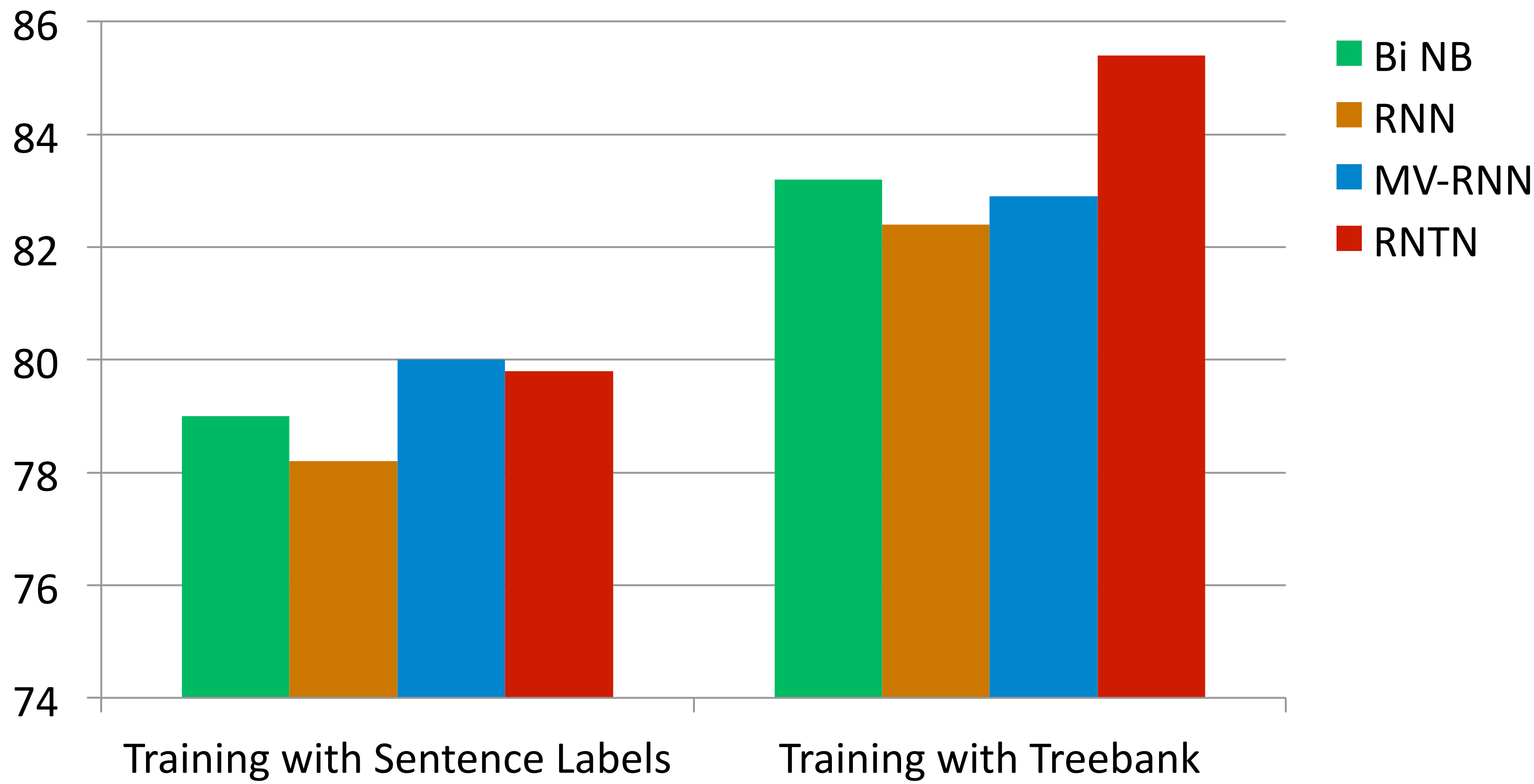


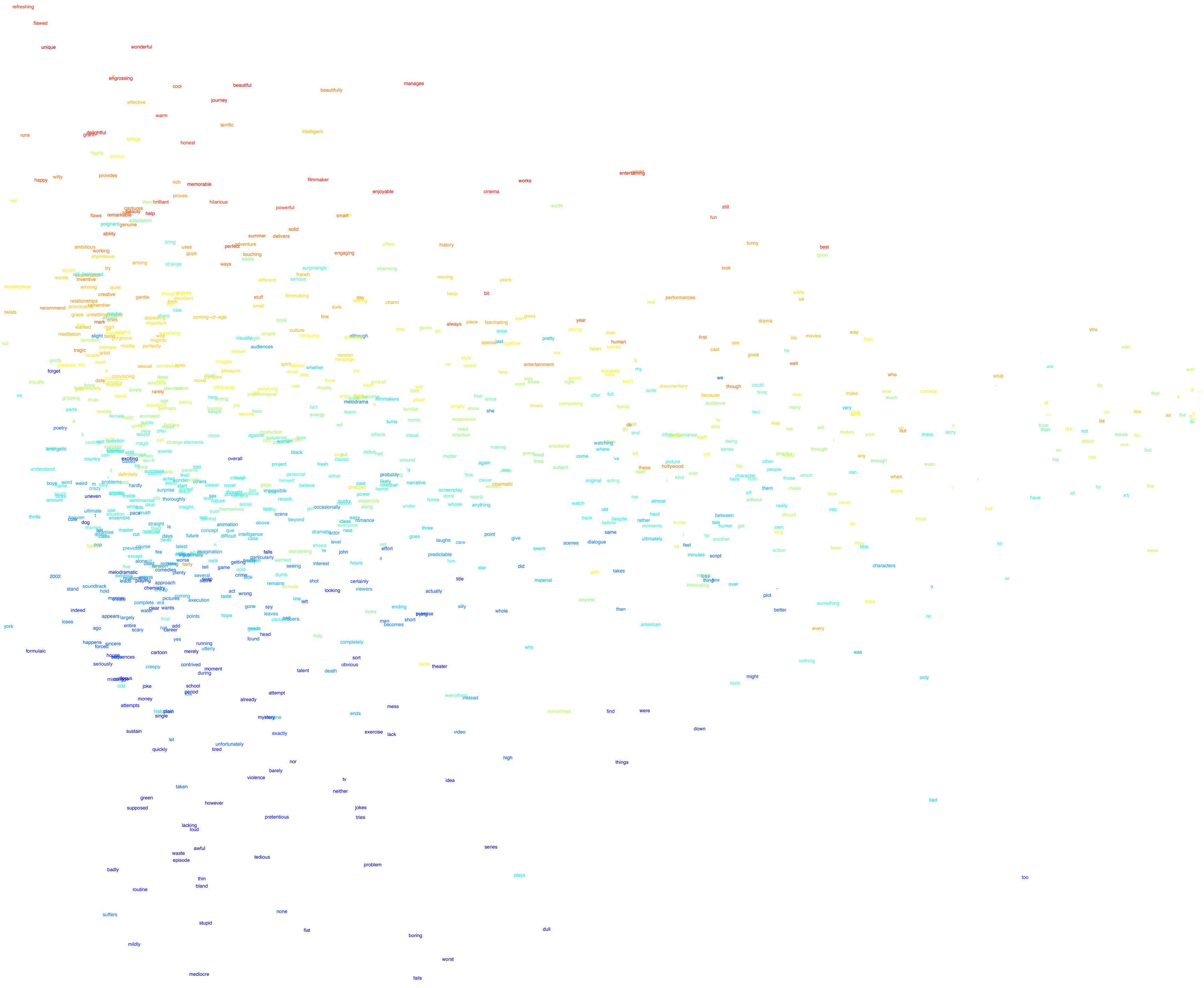
$$p = f\left(\begin{bmatrix}b\\c\end{bmatrix}^T V^{[1:2]} \begin{bmatrix}b\\c\end{bmatrix} + W\begin{bmatrix}b\\c\end{bmatrix}\right)$$



Positive/Negative Results on Treebank

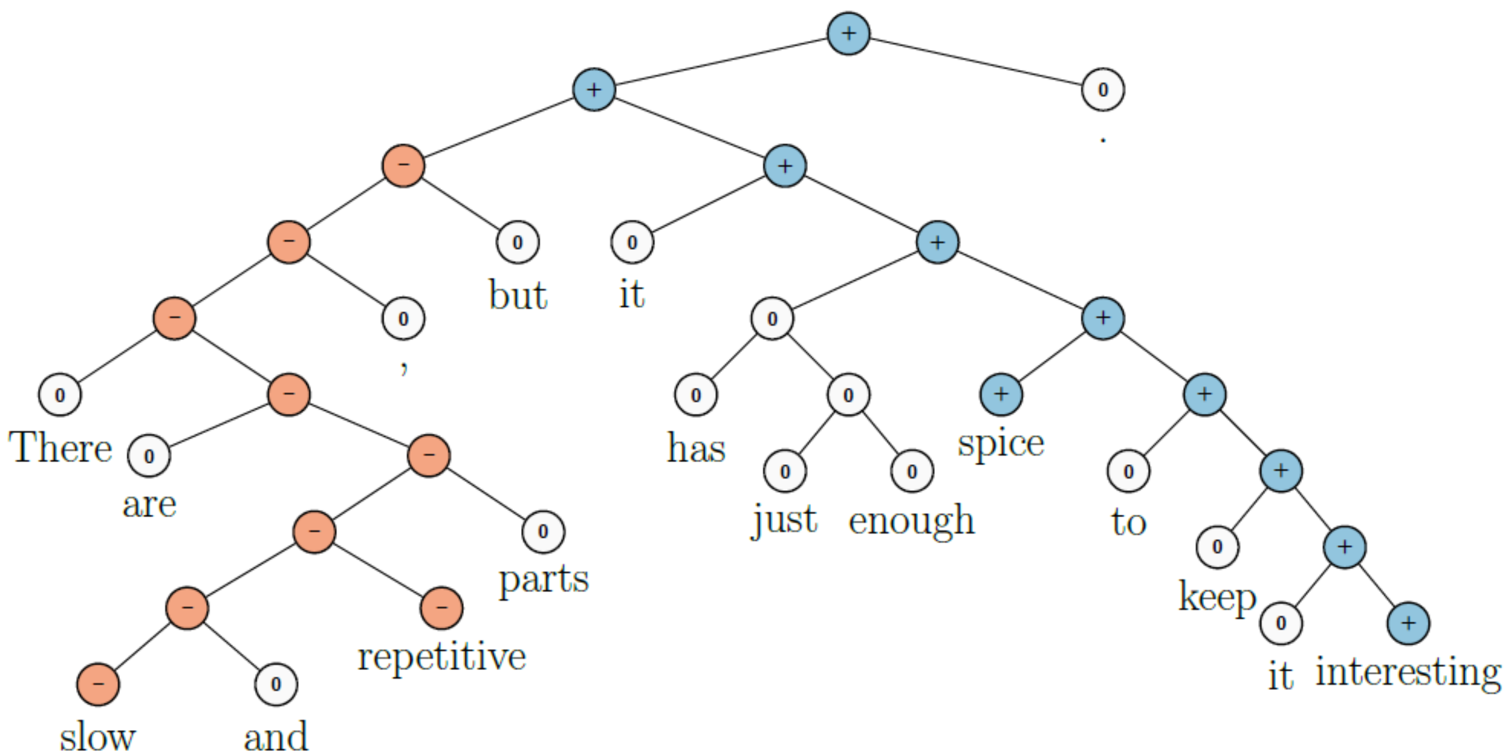
Classifying Sentences: Accuracy improves to 85.4





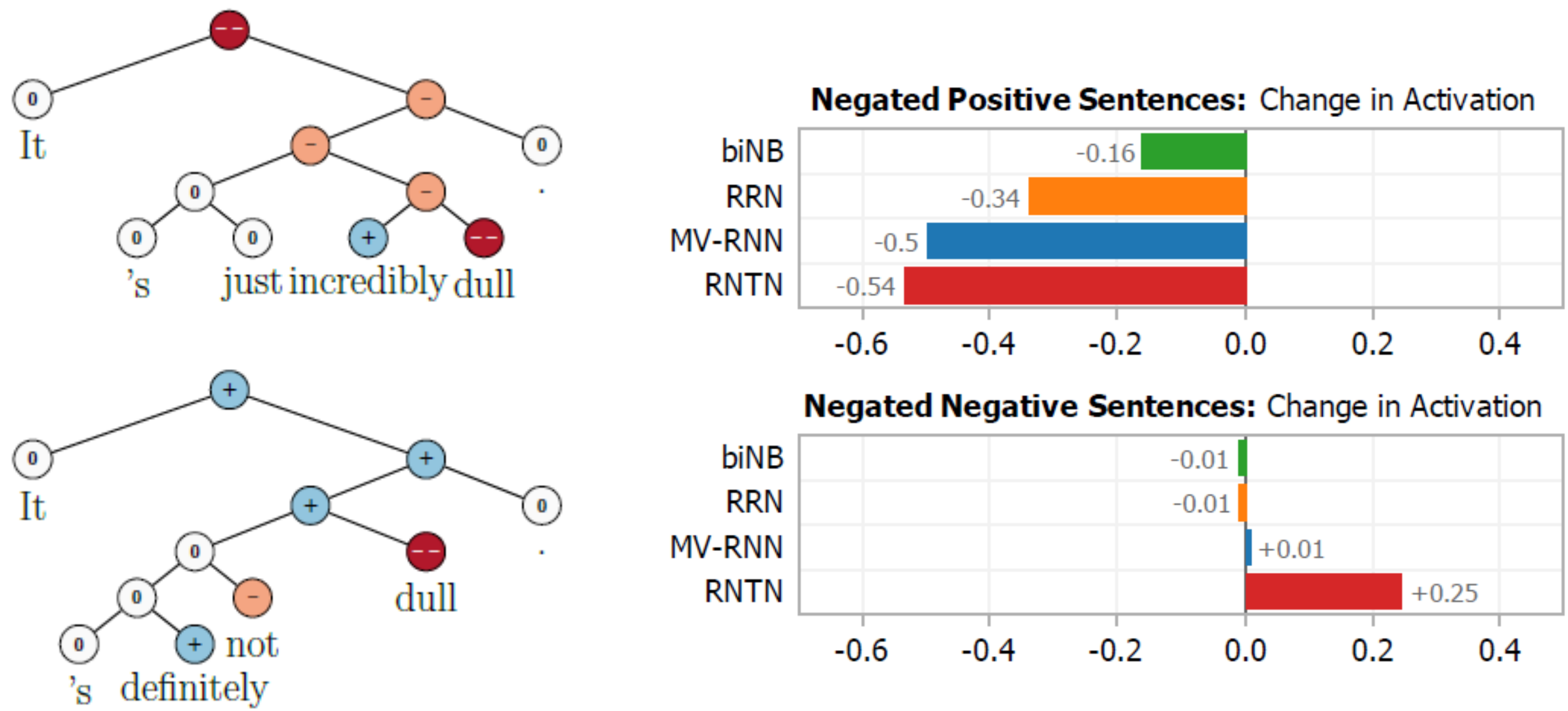
Experimental Results on Treebank

- RNTN can capture constructions like *X but Y*
- RNTN accuracy of 72%, compared to MV-RNN (65%), biword NB (58%) and RNN (54%)



Negation Results

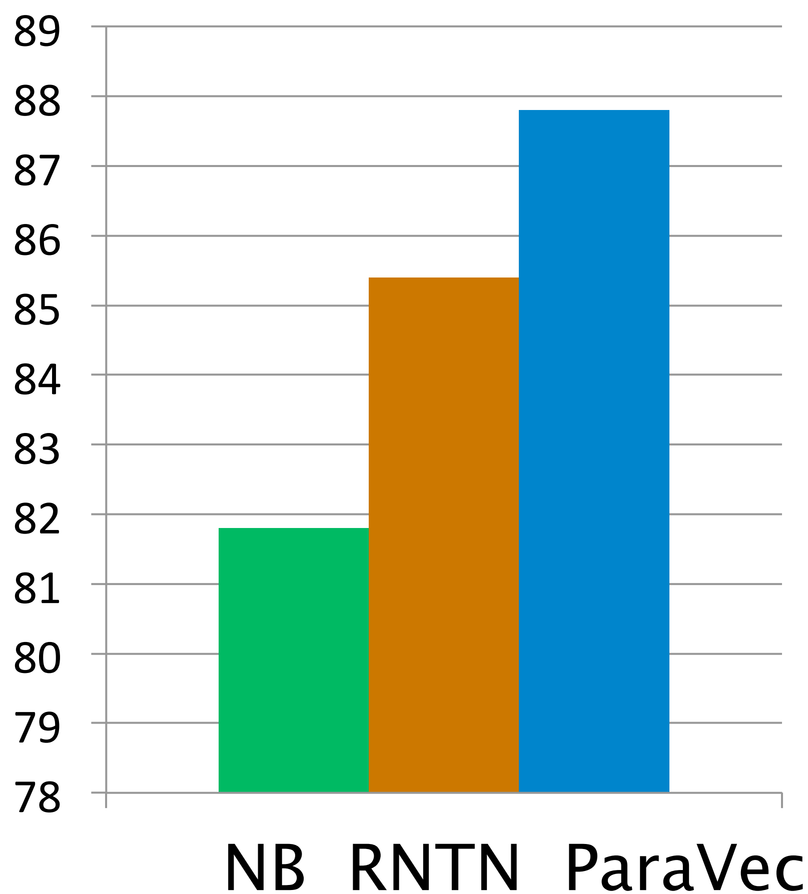
When negating negatives, positive activation should increase!

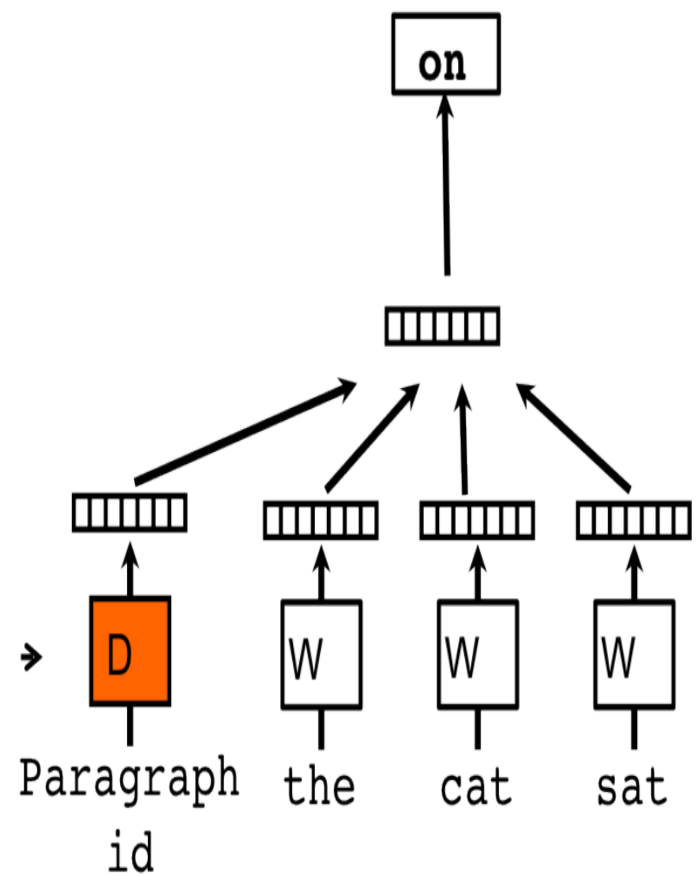


Demo: http://nlp.stanford.edu:8080/sentiment/

A disappointment

Beaten by a **Paragraph Vector** – a word2vec extension with no sentence structure! [Le & Mikolov 2014]





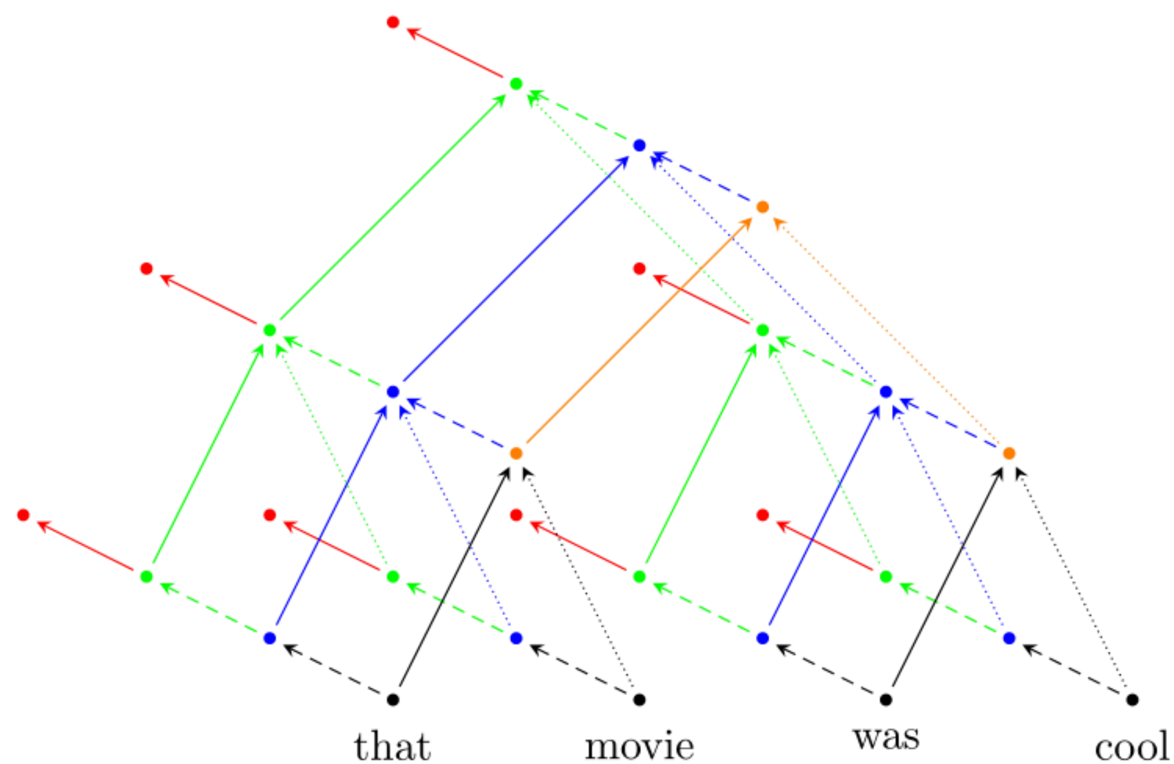
Deep Recursive Neural Networks for Compositionality in Language

(Irsoy & Cardie NIPS, 2014)

Two ideas:

- Separate word and phrase embedding space
- Stack NNs for depth at each node

Beats paragraph vector!



Version 5:
Improving Deep Learning Semantic Representations using a TreeLSTM

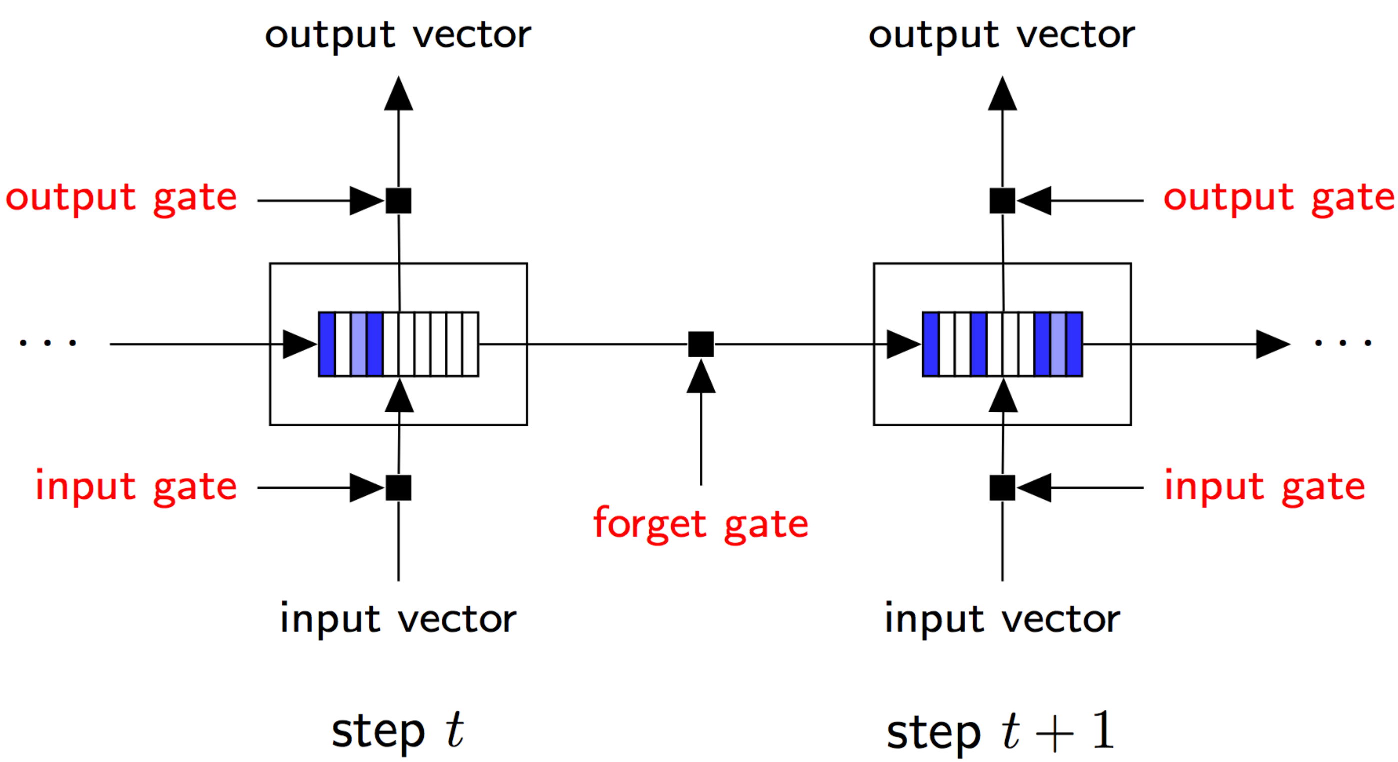
[Tai et al., ACL 2015]

Goals:

- Still trying to represent the meaning of a sentence as a location in a (high-dimensional, continuous) vector space
- In a way that accurately handles semantic composition and sentence meaning
- Generalizing the widely used chain-structured LSTM to trees
- Beat Paragraph Vector!

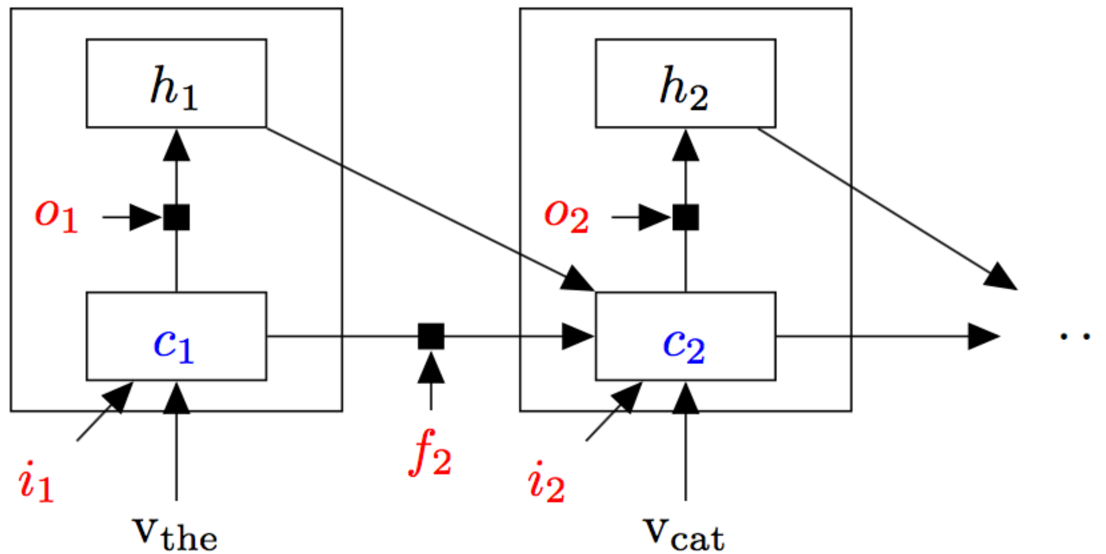
Long Short-Term Memory (LSTM) Units for Sequential Composition

Gates are vectors in $[0,1]^d$ multiplied element-wise for soft masking

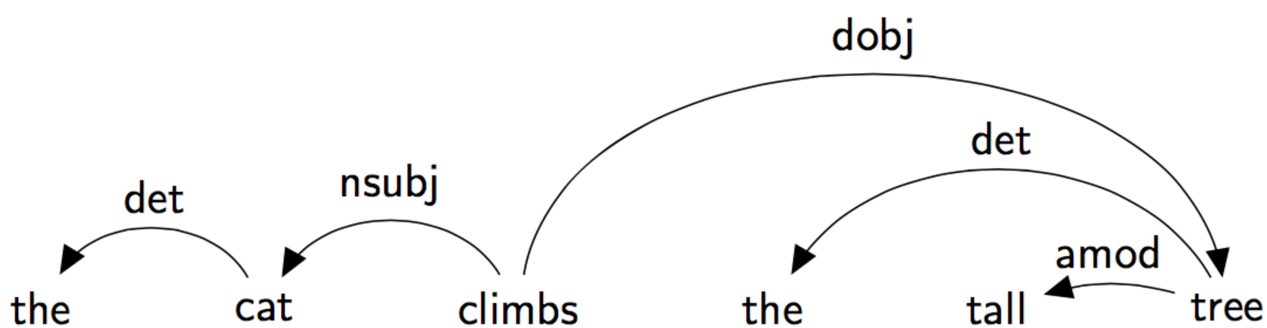


Tree-Structured Long Short-Term Memory Networks

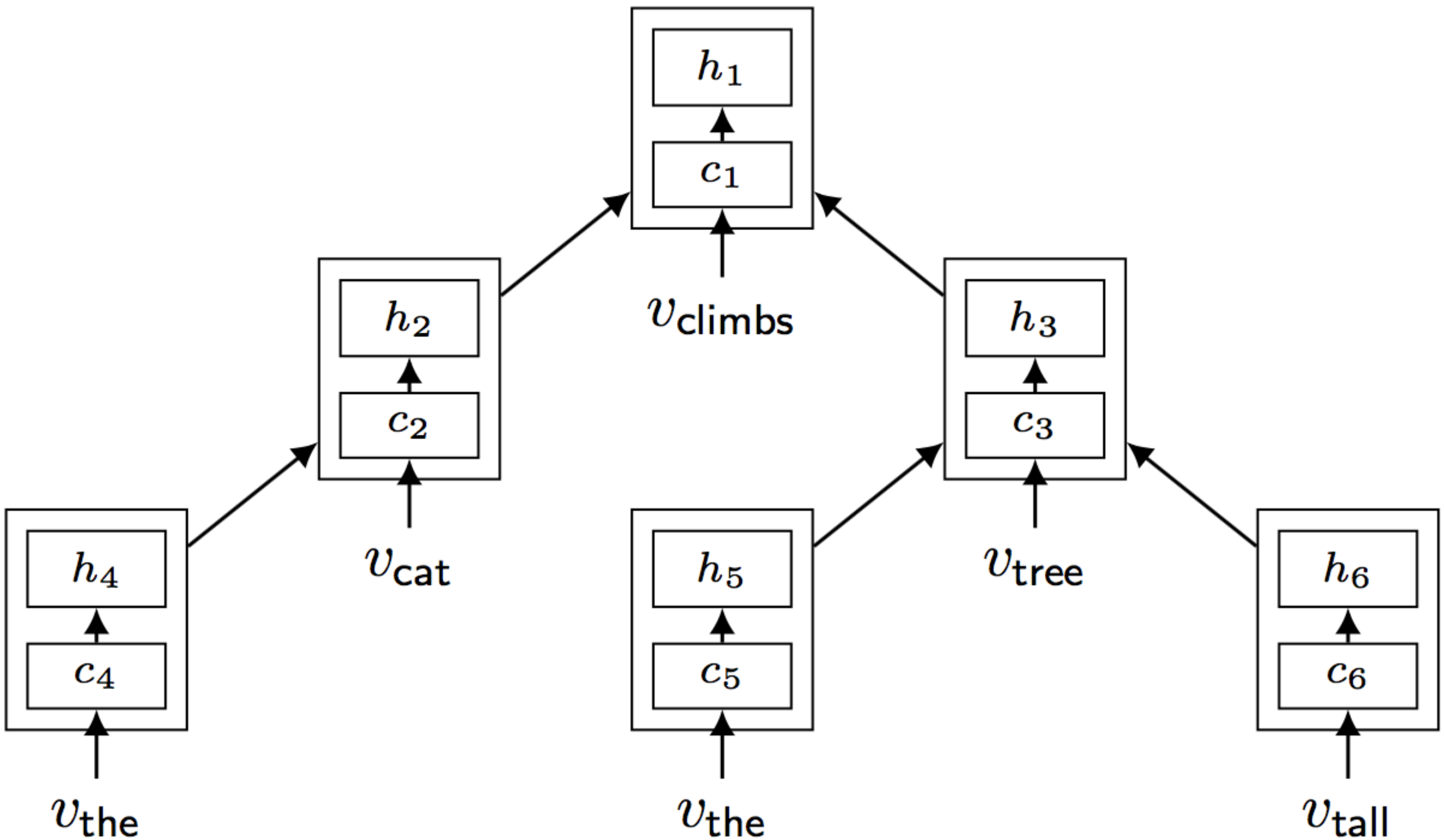
Use Long Short-Term Memories (Hochreiter and Schmidhuber 1997)



Sentences have structure beyond word order – Use this syntactic structure

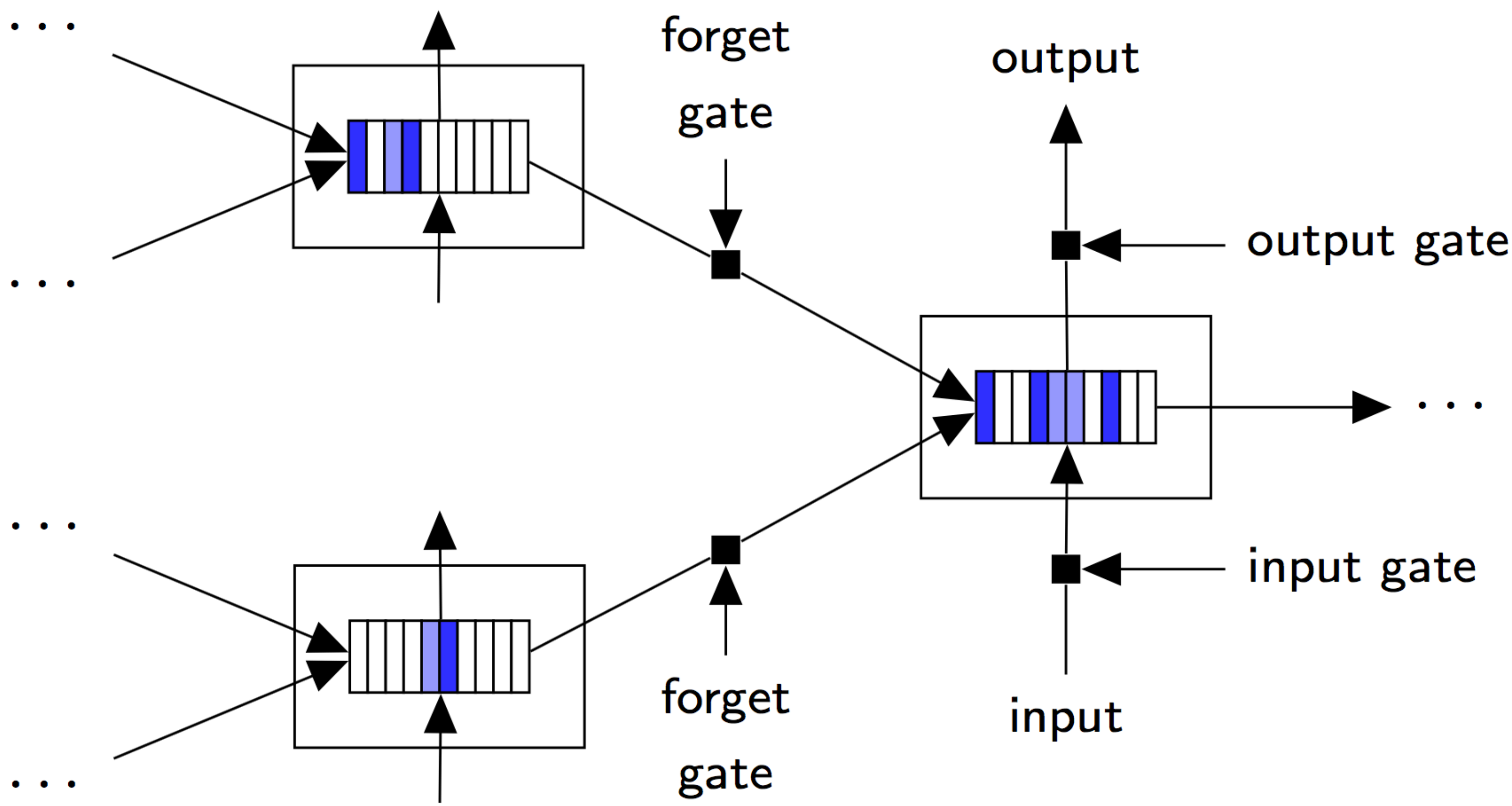


Tree-Structured Long Short-Term Memory Networks
[Tai et al., ACL 2015]



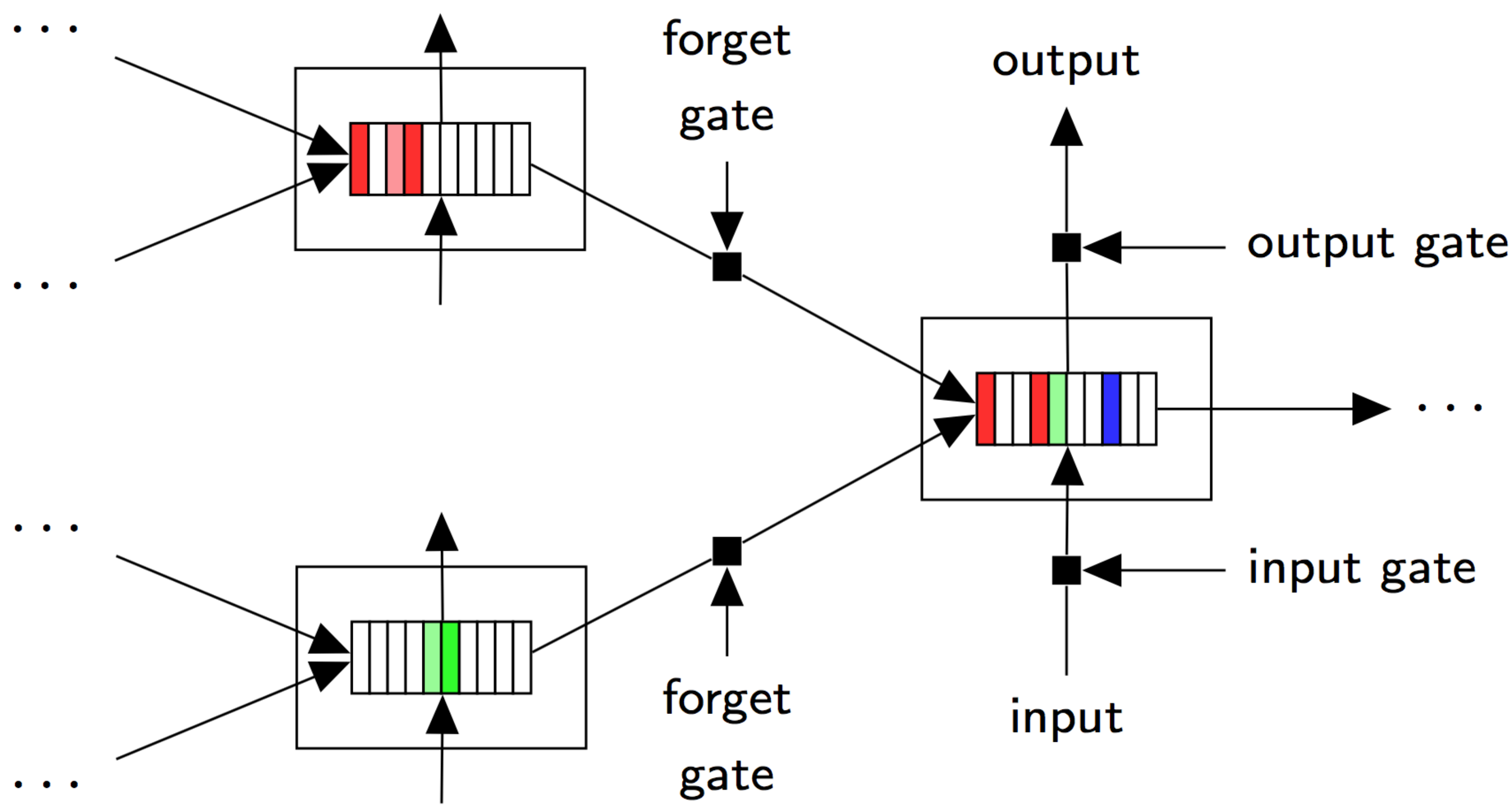
Tree-structured LSTM

Generalizes sequential LSTM to trees with any branching factor



Tree-structured LSTM

Generalizes sequential LSTM to trees with any branching factor



Results: Sentiment Analysis: Stanford Sentiment Treebank

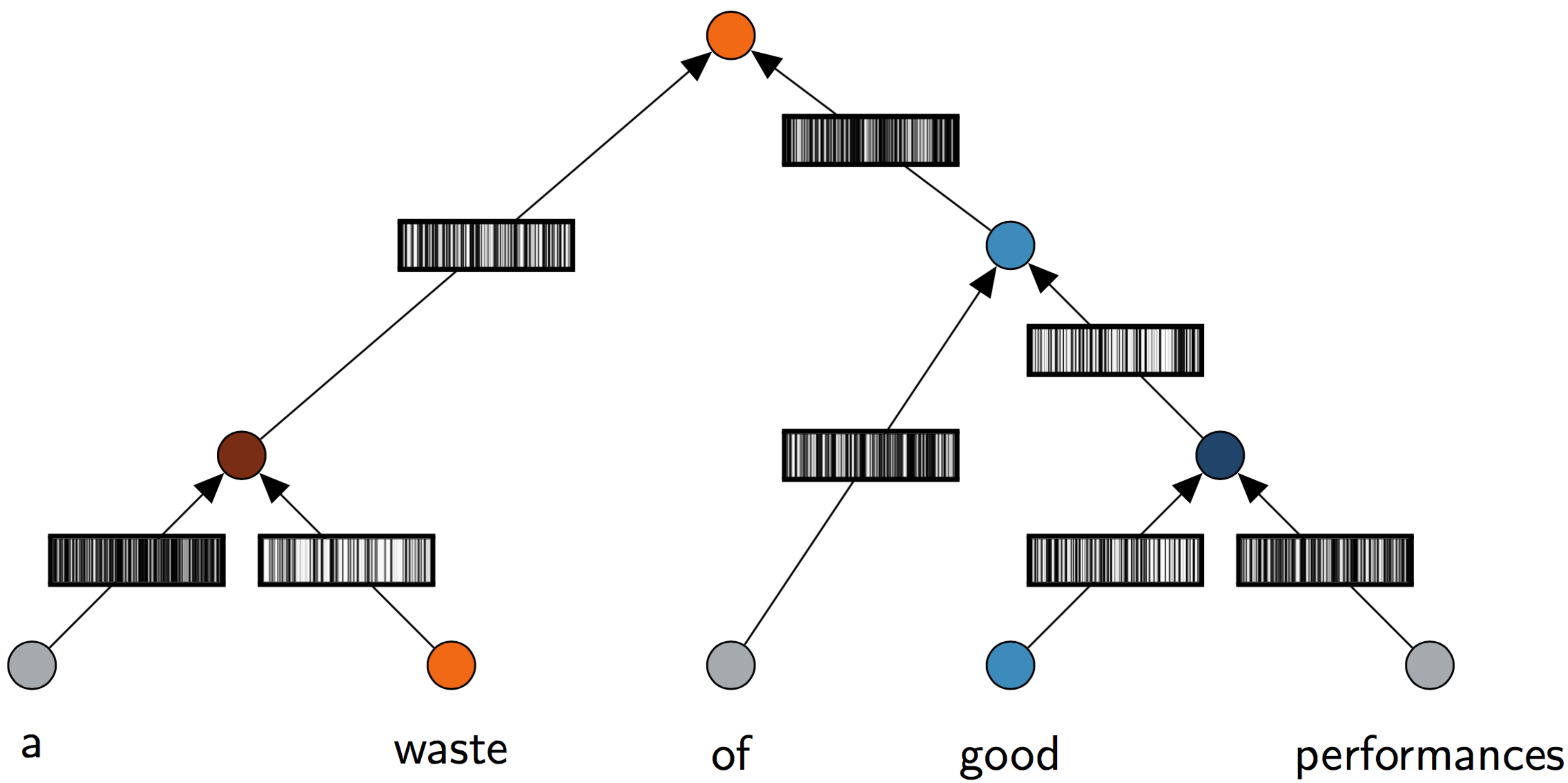
Method	Accuracy % (Fine-grain, 5 classes)
RNTN (Socher et al. 2013)	45.7
Paragraph-Vec (Le & Mikolov 2014)	48.7
DRNN (Irsoy & Cardie 2014)	49.8
LSTM	46.4
Tree LSTM (this work)	**50.9**

Results: Semantic Relatedness
SICK 2014 (Sentences Involving Compositional Knowledge)

Method	Pearson correlation
Word vector average	0.758
Meaning Factory (Bjerva et al. 2014)	0.827
ECNU (Zhao et al. 2014)	0.841
LSTM	0.853
Tree LSTM	**0.868**

Forget Gates: Selective State Preservation

- Stripes = forget gate activations; more white ⇒ more preserved



Tree structure helps

It 's actually pretty good in the first few minutes , but the longer the movie goes , the worse it gets .

Gold	LSTM	TreeLSTM
–	–	–

The longer the movie goes , the worse it gets , but it was actually pretty good in the first few minutes .

Gold	LSTM	TreeLSTM
–	+	–

Natural Language Inference

Can we tell if one piece of text follows from another?

- *Two senators received contributions engineered by lobbyist Jack Abramoff in return for political favors.*
- *Jack Abramoff attempted to bribe two legislators.*

Natural Language Inference = Recognizing Textual Entailment [Dagan 2005, MacCartney & Manning, 2009]

Natural language inference: The 3-way classification task

James Byron Dean refused to move without blue jeans

{**entails**, contradicts, neither}

James Dean didn’t dance without pants

The task: Natural language inference

Claim: Simple task to define, but engages the full complexity of compositional semantics:

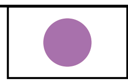
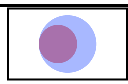
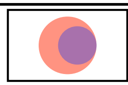
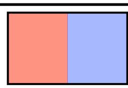
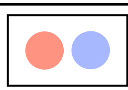
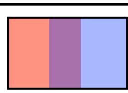
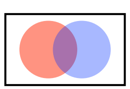
- Lexical entailment
- Quantification
- Coreference
- Lexical/scope ambiguity
- Commonsense knowledge
- Propositional attitudes
- Modality
- Factivity and implicativity

  ...

Natural Logic approach: relations

(van Benthem 1988, MacCartney & Manning 2008)

Seven possible relations between phrases/sentences:

	$x \equiv y$	equivalence	*couch* ≡ *sofa*
	$x \sqsubset y$	forward entailment (strict)	*crow* ⊏ *bird*
	$x \sqsupset y$	reverse entailment (strict)	*European* ⊐ *French*
	$x \,^\wedge\, y$	negation (exhaustive exclusion)	*human* ^ *nonhuman*
	$x \mid y$	alternation (non-exhaustive exclusion)	*cat* | *dog*
	$x \smile y$	cover (exhaustive non-exclusion)	*animal* ‿ *nonhuman*
	$x \# y$	independence	*hungry* # *hippo*

Natural logic: relation joins

	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\equiv$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\sqsubset$	$\sqsubset$	$\sqsubset$	$\cdot$	$\mid$	$\mid$	$\cdot$	$\cdot$
$\sqsupset$	$\sqsupset$	$\cdot$	$\sqsupset$	$\smile$	$\cdot$	$\smile$	$\cdot$
$\wedge$	$\wedge$	$\smile$	$\mid$	$\equiv$	$\sqsupset$	$\sqsubset$	$\#$
$\mid$	$\mid$	$\cdot$	$\mid$	$\sqsubset$	$\cdot$	$\sqsubset$	$\cdot$
$\smile$	$\smile$	$\smile$	$\cdot$	$\sqsupset$	$\sqsupset$	$\cdot$	$\cdot$
$\#$	$\#$	$\cdot$	$\cdot$	$\#$	$\cdot$	$\cdot$	$\cdot$

Can our NNs learn to make these inferences over pairs of embedding vectors?

MacCartney's natural Logic

An implementable logic for natural language inference without logical forms. (MacCartney and Manning '09)

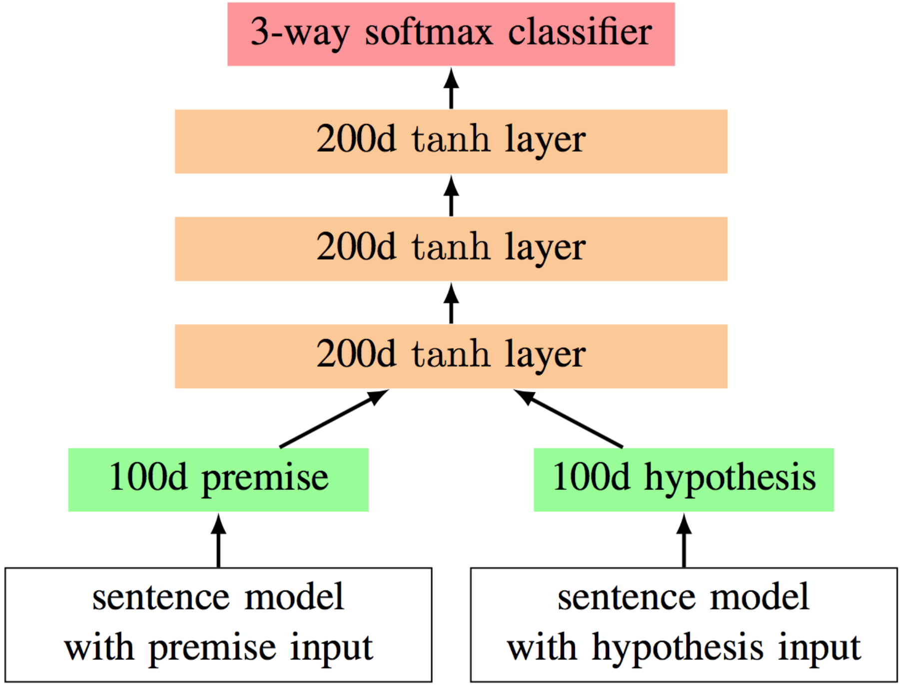
- Sound logical interpretation (Icard and Moss '13)

P	*James Dean*	*refused to*			*move*	*without*	*blue*	*jeans*
H	*James Byron Dean*		*did*	*n't*	*dance*	*without*		*pants*
edit index	1	2	3	4	5	6	7	8
edit type	SUB	DEL	INS	INS	SUB	MAT	DEL	SUB
lex feats	strsim= 0.67	implic: –/o	cat:aux	cat:neg	hypo			hyper
lex entrel	=	|	=	^	⊐	=	⊏	⊏
projectivity	↑	↑	↑	↑	↓	↓	↑	↑
atomic entrel	=	|	=	^	⊏	=	⊏	⊏

inversion

A neural network for NLI
[Bowman 2014]

- Words are learned embedding vectors.
- One TreeRNN or TreeRNTN per sentence
- Softmax emits label
- Learn everything with SGD.



Natural language inference data

[Bowman, Manning & Potts, to appear EMNLP 2015]

- To do NLI on real English, we need to teach an NN model English almost from scratch
- What data do we have to work with:
  - **Word embeddings**: GloVe/word2vec (useful with any data source)
  - **SICK**: Thousands of examples created by editing and pairing hundreds of sentences
  - **RTE**: Hundreds of examples created by hand
  - **DenotationGraph**: Millions of extremely noisy examples (~73% correct?) constructed fully automatically

Results on SICK (+DG, +tricks)

	SICK Train	DG Train	Test
Most freq. class	**56.7%**	**50.0%**	**56.7%**
30 dim TreeRNN	95.4%	67.0%	74.9%
50 dim TreeRNTN	97.8%	74.0%	**76.9%**

Are we competitive on SICK? Sort of...

Best result (U. Illinois)	**84.5%**
≈ interannotator agreement!	
Median submission (out of 18):	77%
Our TreeRNTN:	76.9%

We're a purely-learned system
None of the ones in the competition were

Natural language inference data

[Bowman, Manning & Potts, to appear EMNLP 2015]

- To do NLI on real English, we need to teach an NN model English almost from scratch
- What data do we have to work with:
  - GloVe/word2vec (useful w/ any data source)
  - SICK: Thousands of examples created by editing and pairing hundreds of sentences
  - RTE: Hundreds of examples created by hand
  - DenotationGraph: Millions of extremely noisy examples (~73% correct?) constructed fully automatically
  - **Stanford NLI corpus: ~600k examples, written by Turkers**

The Stanford NLI corpus

Instructions

The Stanford University NLP Group is collecting data for use in research on computer understanding of English. We appreciate your help!

We will show you the caption for a photo. We will not show you the photo. Using only the caption and what you know about the world:

- Write one alternate caption that is **definitely** a **true** description of the photo.
- Write one alternate caption that **might be** a **true** description of the photo.
- Write one alternate caption that is **definitely** an **false** description of the photo.

Photo caption **A little boy in an apron helps his mother cook.**

Definitely correct Example: For the caption *"Two dogs are running through a field."* you could write *"There are animals outdoors."*

Write a sentence that follows from the given caption.

Maybe correct Example: For the caption *"Two dogs are running through a field."* you could write *"Some puppies are running to catch a stick."*

Write a sentence which may be true given the caption, and may not be.

Definitely incorrect Example: For the caption *"Two dogs are running through a field."* you could write *"The pets are sitting on a couch."*

Write a sentence which contradicts the caption.

Problems (optional) *If something is wrong with the caption that makes it difficult to understand, do your best above and let us know here.*

Initial SNLI Results

Model	Accuracy
100d sum of words	75.3
100d TreeRNN	72.2
100d LSTM TreeRNN	77.6

Envoi

We want more than word meanings!

We want:

- Meanings of larger units, calculated compositionally
- The ability to do natural language inference