Jamie Mutton
@jcmm33

Follow

If in doubt, fire up the Stanford Parser to help with the kids homework :-)

11/6/13, 11:46 AM

3 RETWEETS

Recursive Deep Learning for Modeling Semantic Compositionality

Christopher Manning

Stanford University

Richard Socher

&

John Bauer, Danqi Chen, Jason Chuang, Eric Huang,
Brody Huval, Cliff Lin, Minh-Thang Luong, Andrew Ng,
Jeffrey Pennington, Alex Perelygin, Christopher Potts, & Jean Wu

Neural word embeddings are great for NLP!

Denser (semantic) word representations are great!

Word representations built from contexts of use capture meaning similarity and greatly help all NLP applications

- +1.4% F1 Dependency Parsing

  15.2% error reduction (Koo & Collins 2008)

- +3.4% F1 Named Entity Recognition

  23.7% error reduction (Stanford NER, exchange clustering)

"You shall know a word by the company it keeps"

(J. R. Firth 1957: 11)

But what next?

Great new work at this conference:

- Andriy Mnih and Koray Kavukcuoglu
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean

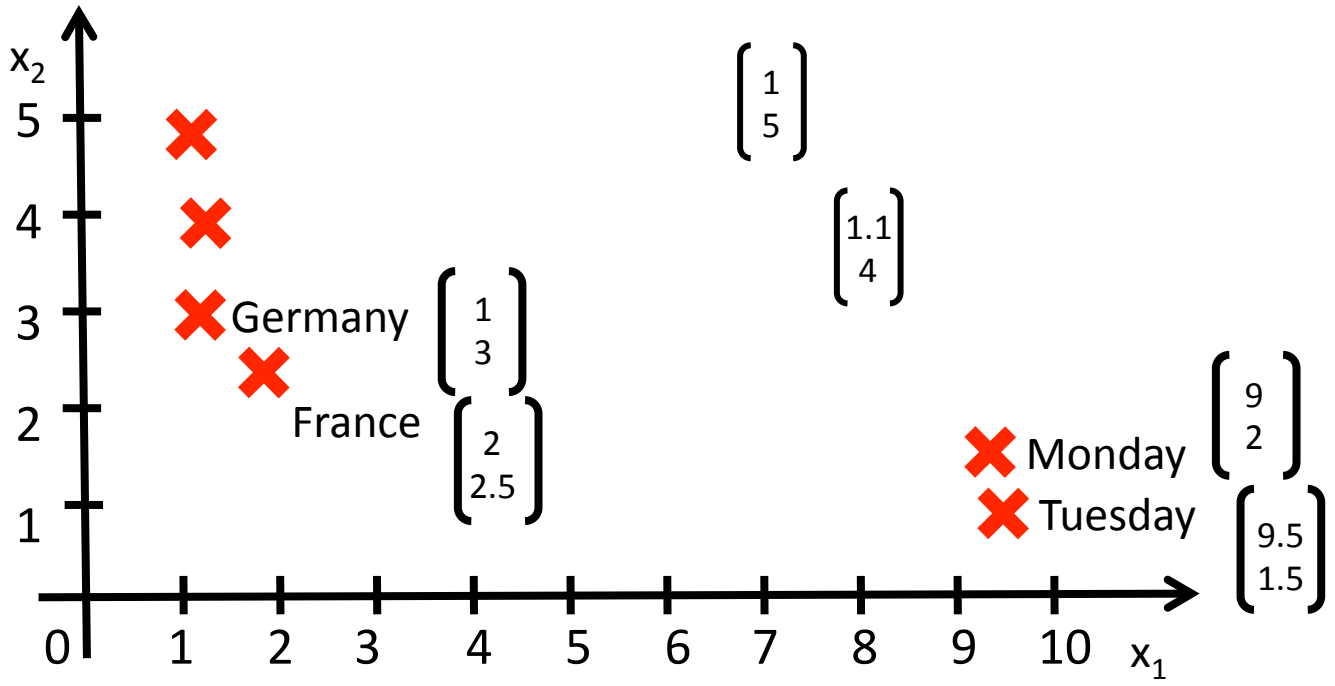
But we need more! What of larger semantic units?

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

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

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

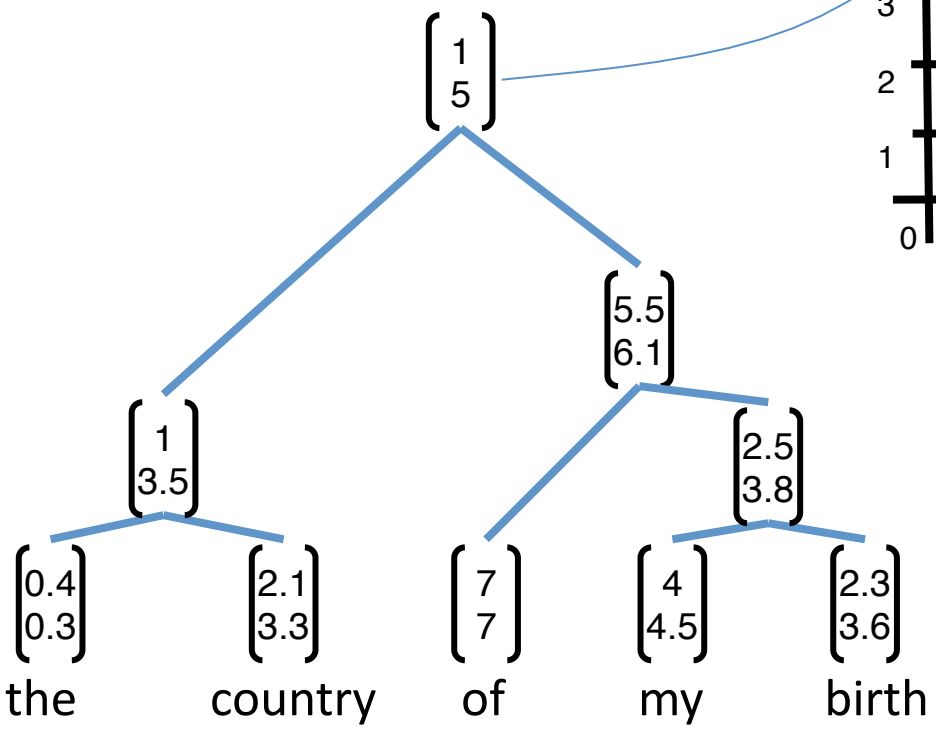
the place where I was born

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

How should we map phrases into a vector space?

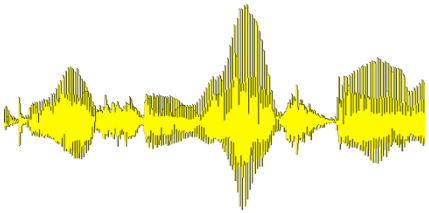
Use the principle of compositionality!

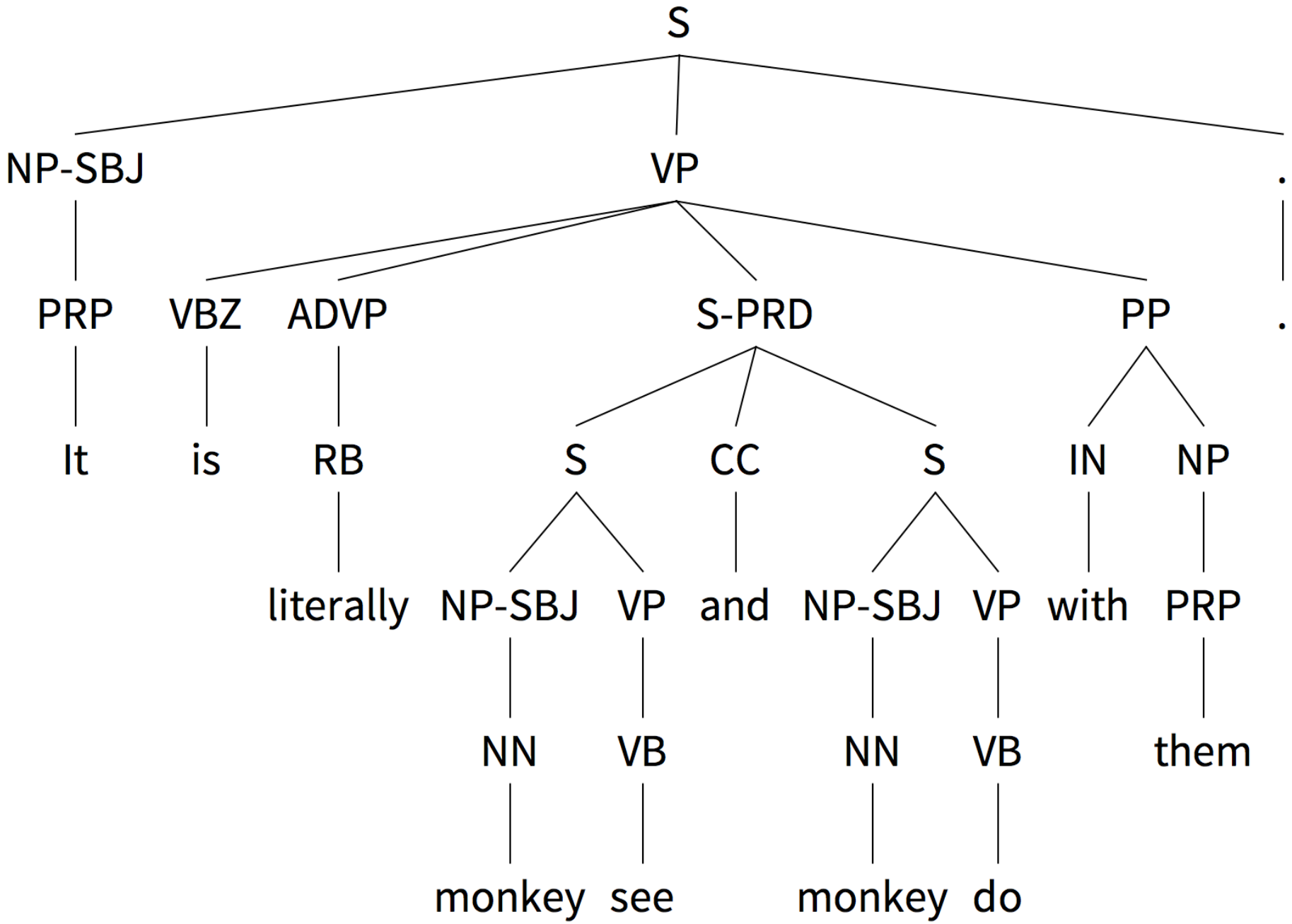
The meaning (vector) of a sentence is determined by
(1) the meanings of its words and
(2) a method that combine them.

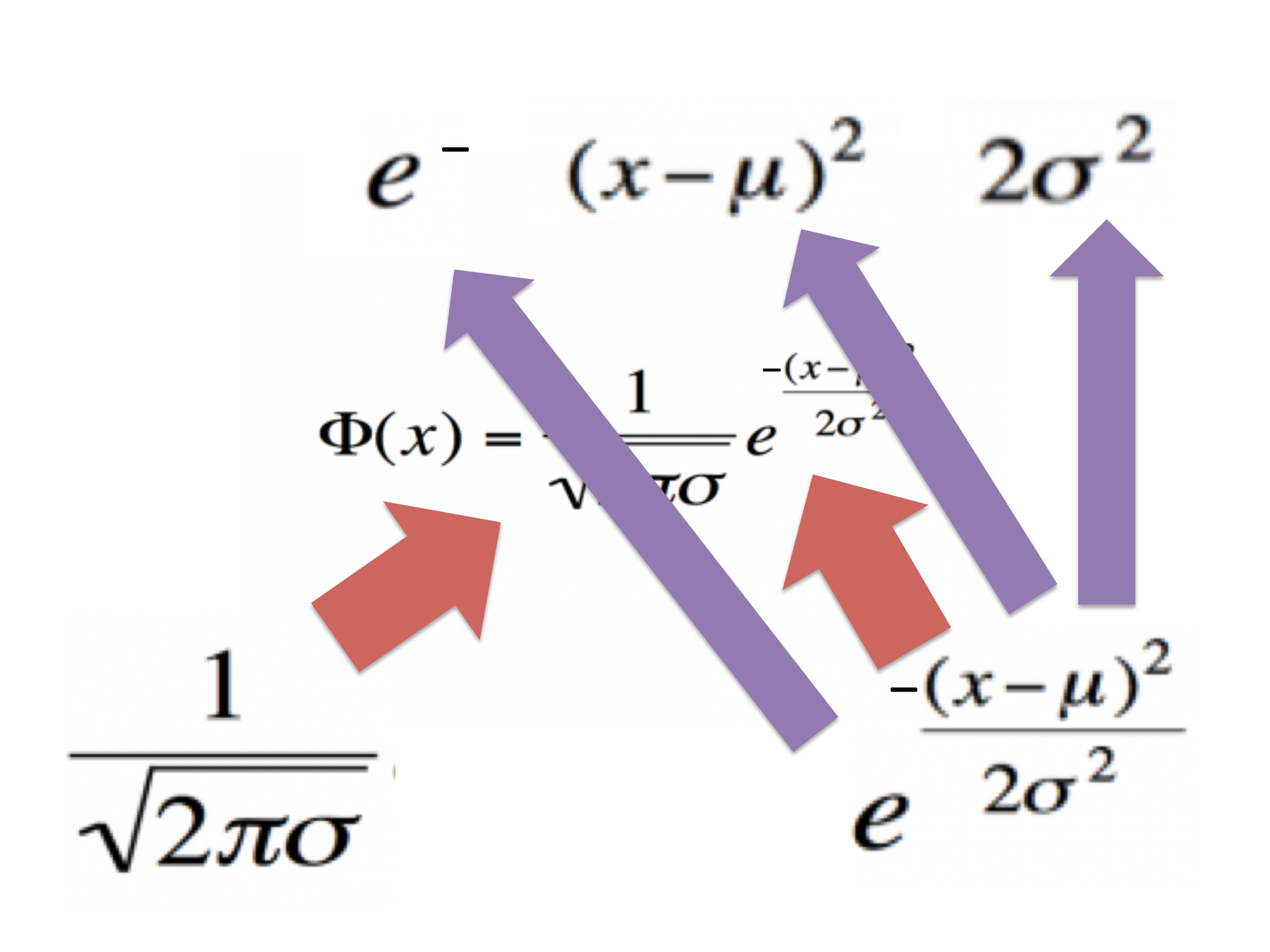


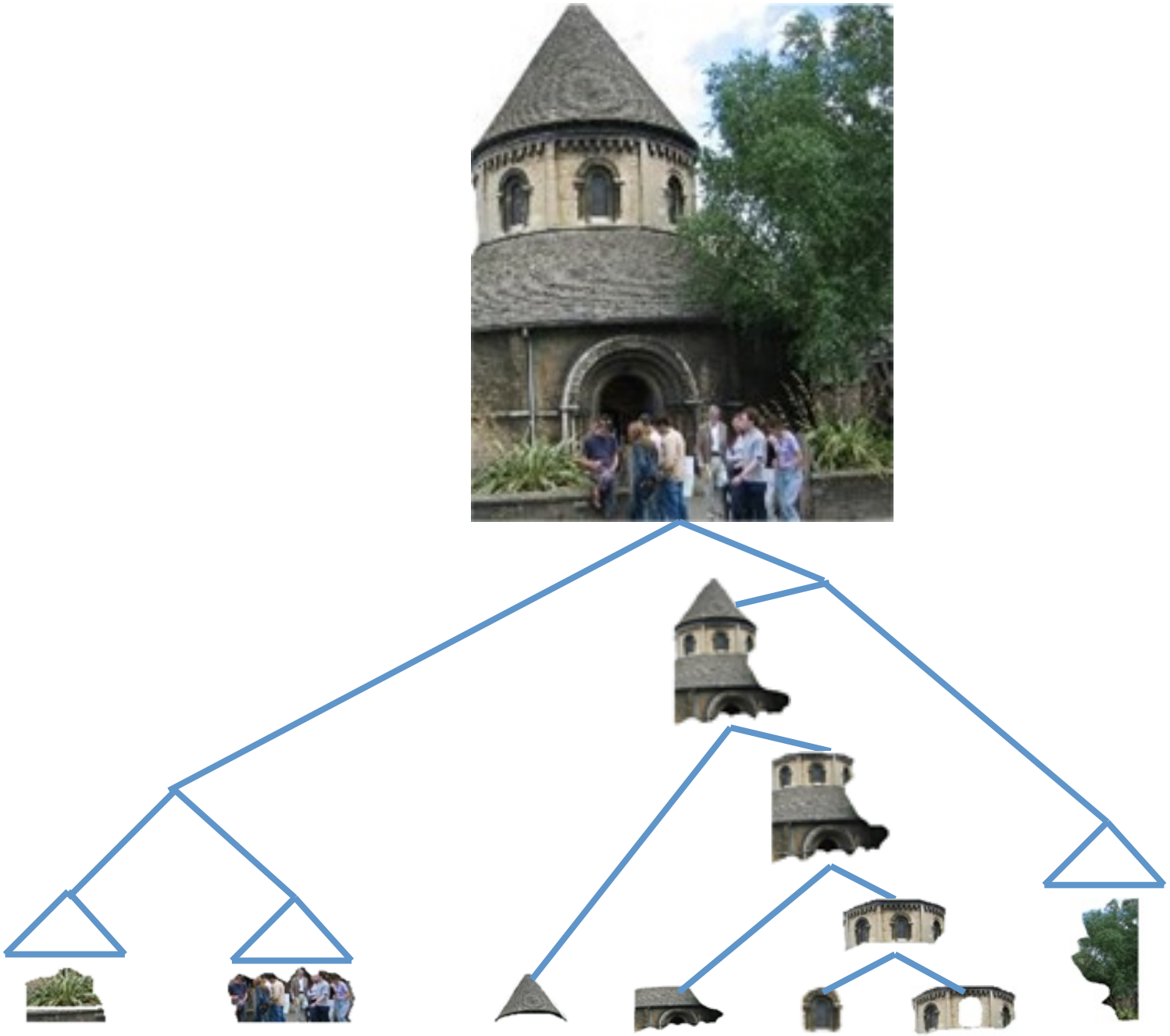
Human language

- Deep neural networks have been very successful in unsupervised feature learning over sensory inputs
  - Audio, vision
- But the representations used and learned are
  - Dense, Flat, Fixed size with Layers Uniformly and Densely connected
- Human languages differ
  - Languages can use sound, images (writing), or gesture as a substrate
  - But above these is a symbolic system
  - Meaning is represented by hierarchical composition of pieces



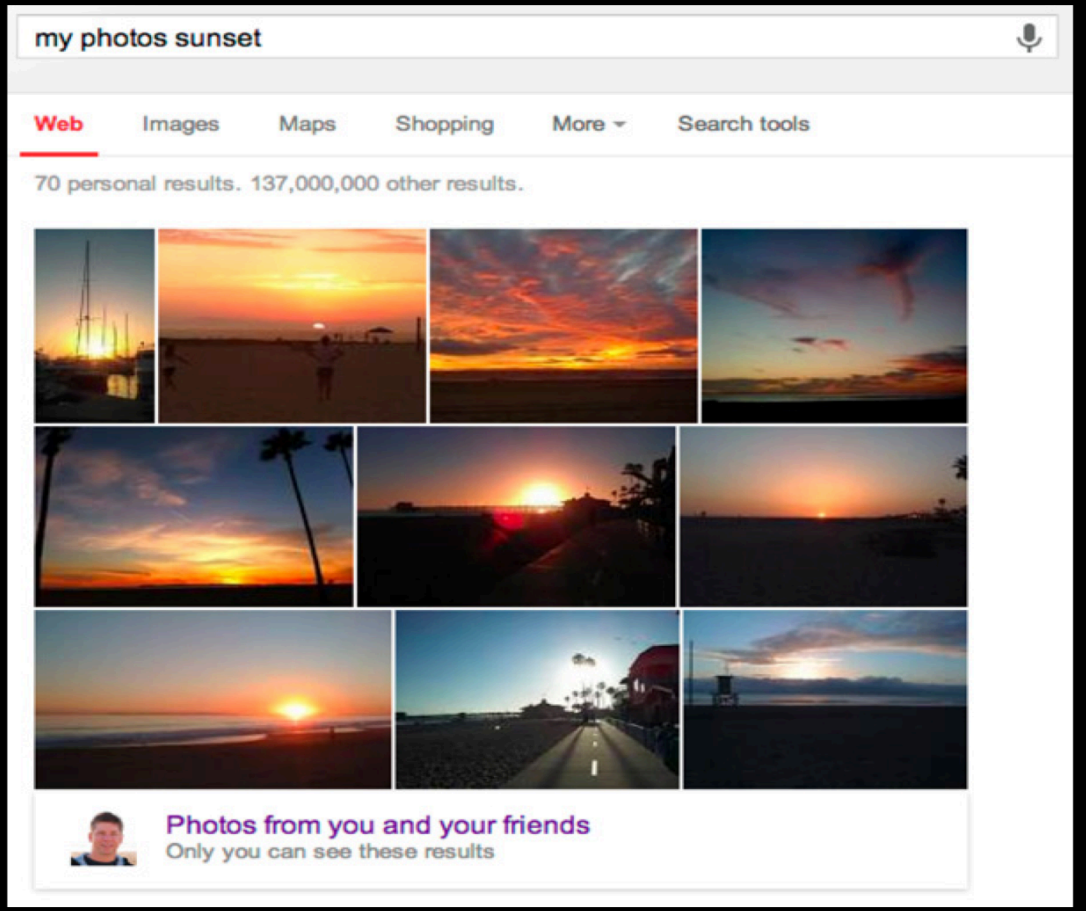

S
NP-SBJ
VP
.
PRP
VBZ
ADVP
S-PRD
PP
.
It
is
RB
S
CC
S
IN
NP
literally
NP-SBJ
VP
and
NP-SBJ
VP
with
PRP
NN
VB
NN
VB
them
monkey
see
monkey
do


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



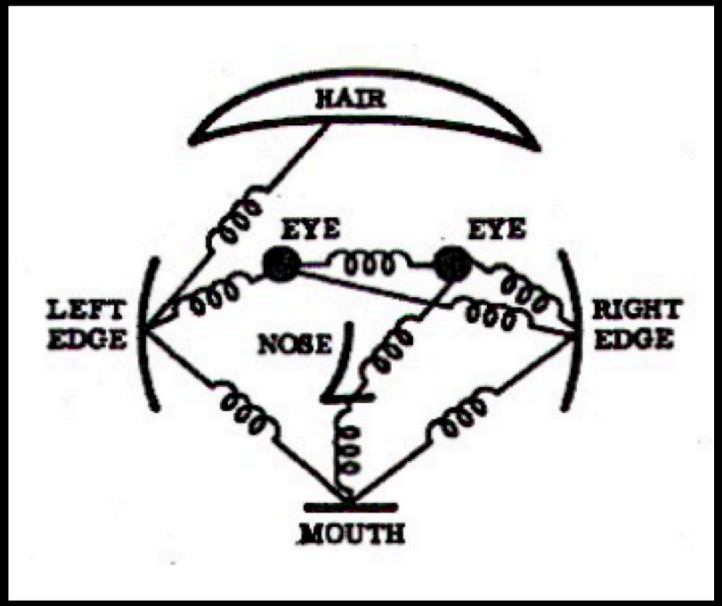
Convolutional Neural Networks

- Google & Baidu, Spring 2013 for personal image search



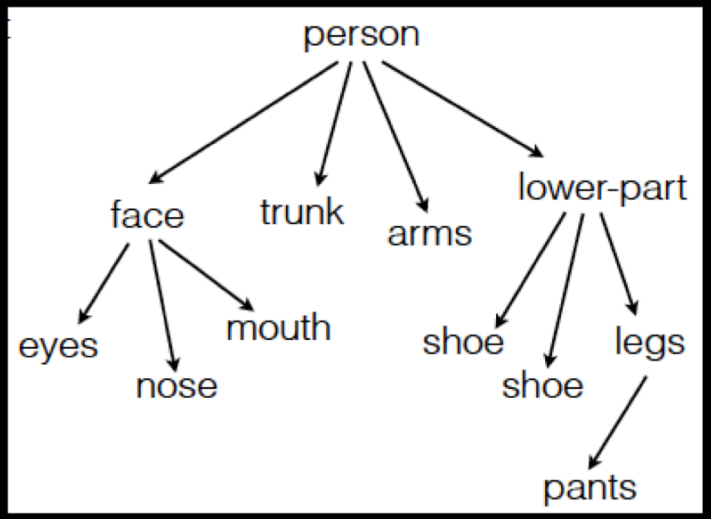
Deep Learning + Structured Prediction

- ConvNet feature extractor
- Combine with top-down reasoning



[Fischler and R. Elschlager 1973]

Stochastic Grammars



[R. Girshick, P. Felzenszwalb, D. McAllester, Object Detection with Grammar Models, NIPS 2011]

Artificial Intelligence is not primarily about performing classifications of sensory inputs

Artificial Intelligence is largely about understanding bigger things from knowing about smaller things

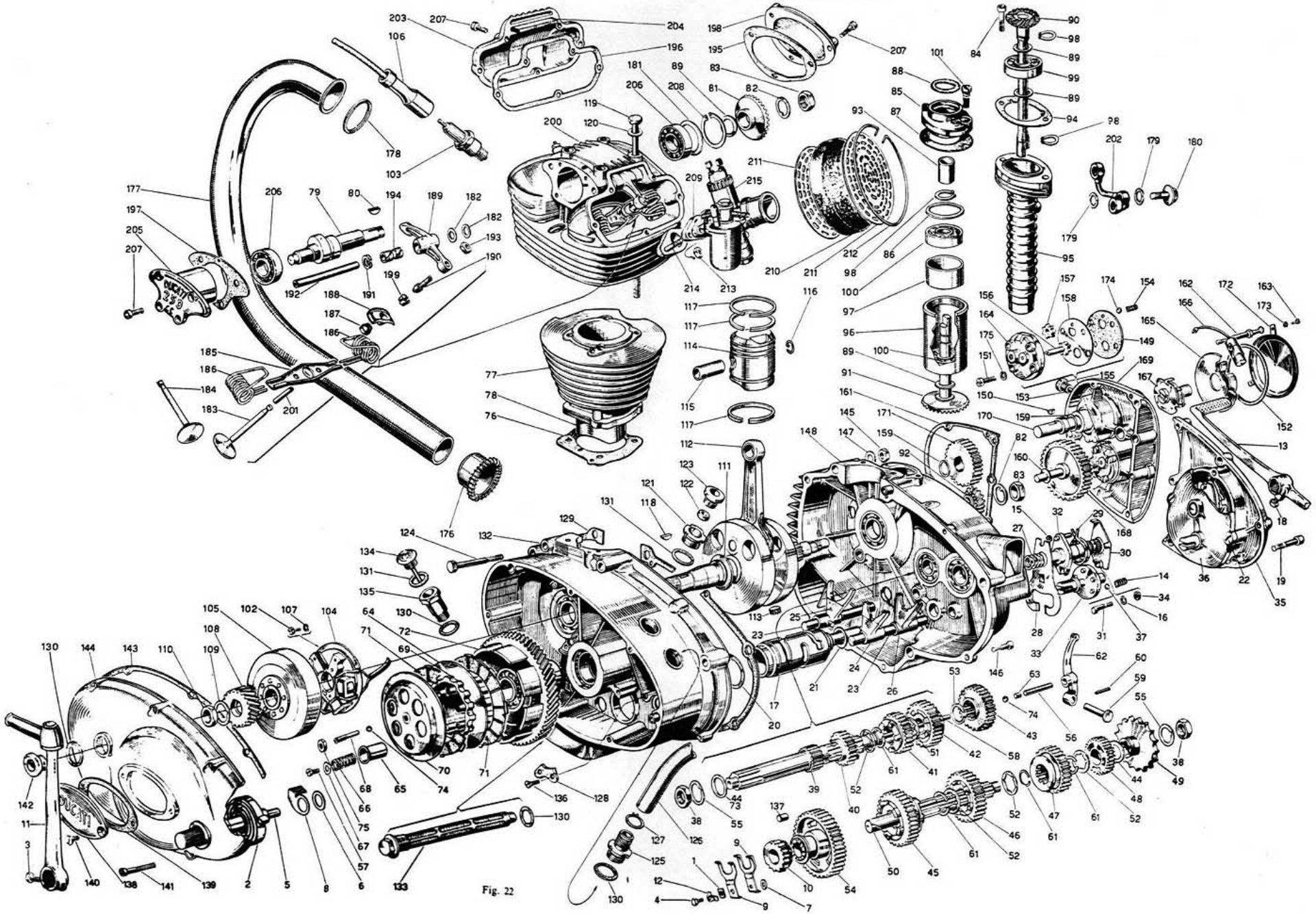


Fig. 22

How can we build

meaning composition functions

in deep learning systems?

Menu

1. Wanted: meaning composition functions
2. Recursive Neural Networks
3. Parsing with Compositional Vector Grammars
4. Relation extraction with Matrix-Vector Networks
5. Sentiment with Recursive Neural Tensor Networks

Recursive Neural Networks for Phrase Vectors

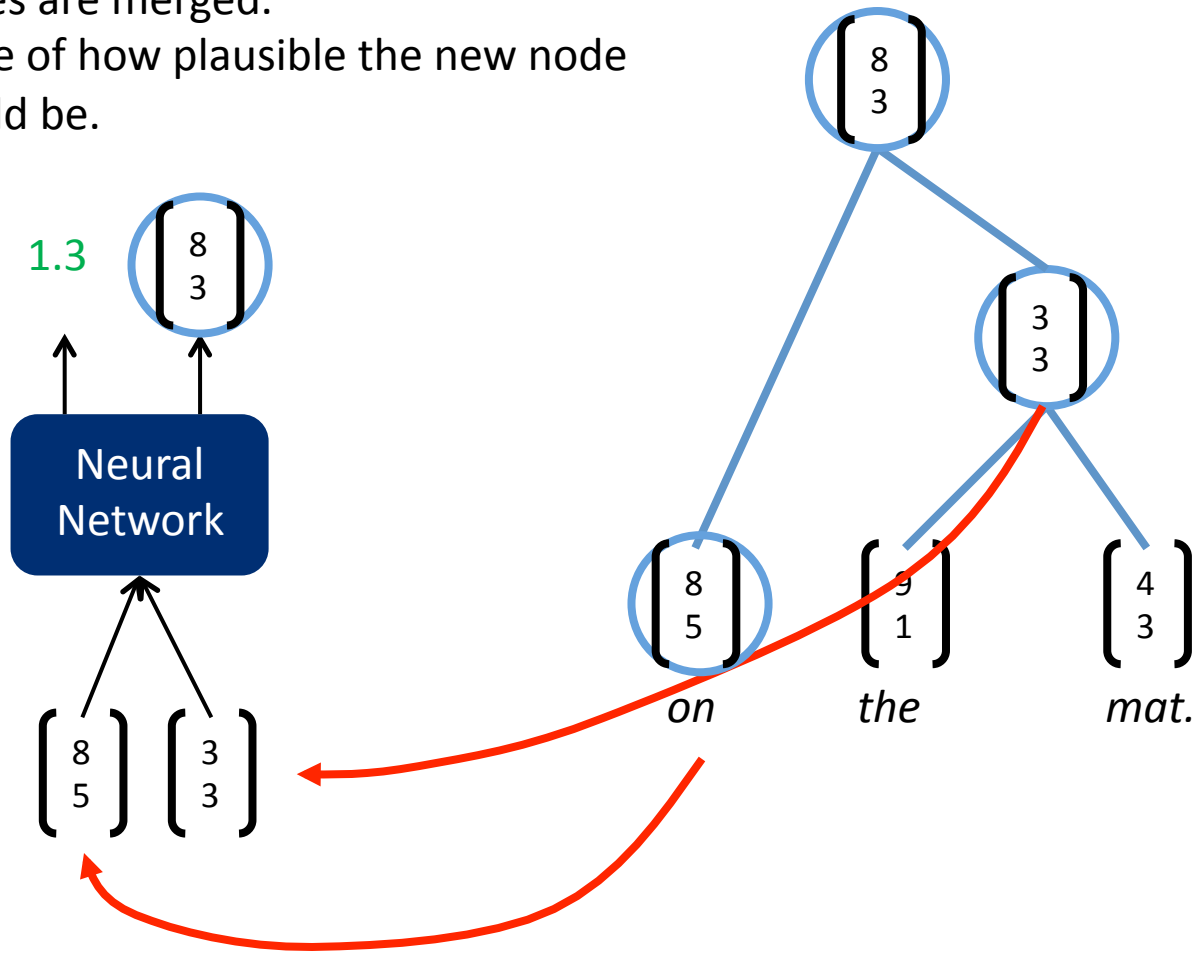
Basic computational unit: Recursive Neural Network

Socher et al. (ICML, 2011)

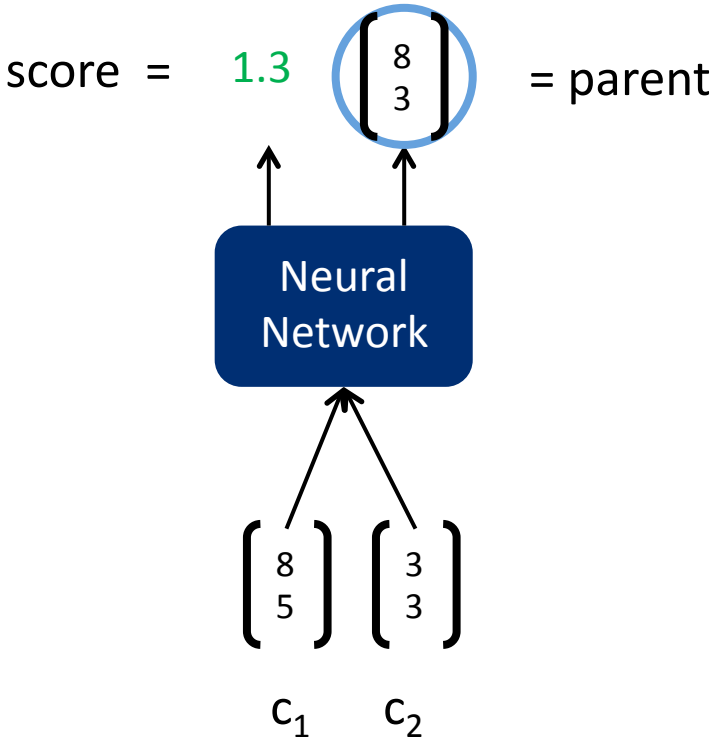
Inputs: two candidate children's representations

Outputs:

1. The semantic representation of the two nodes are merged.
2. Score of how plausible the new node would be.



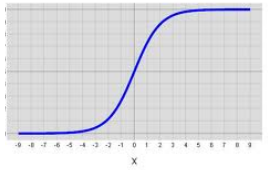
Recursive Neural Network Definition



$$\text{score} = V^T p$$

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

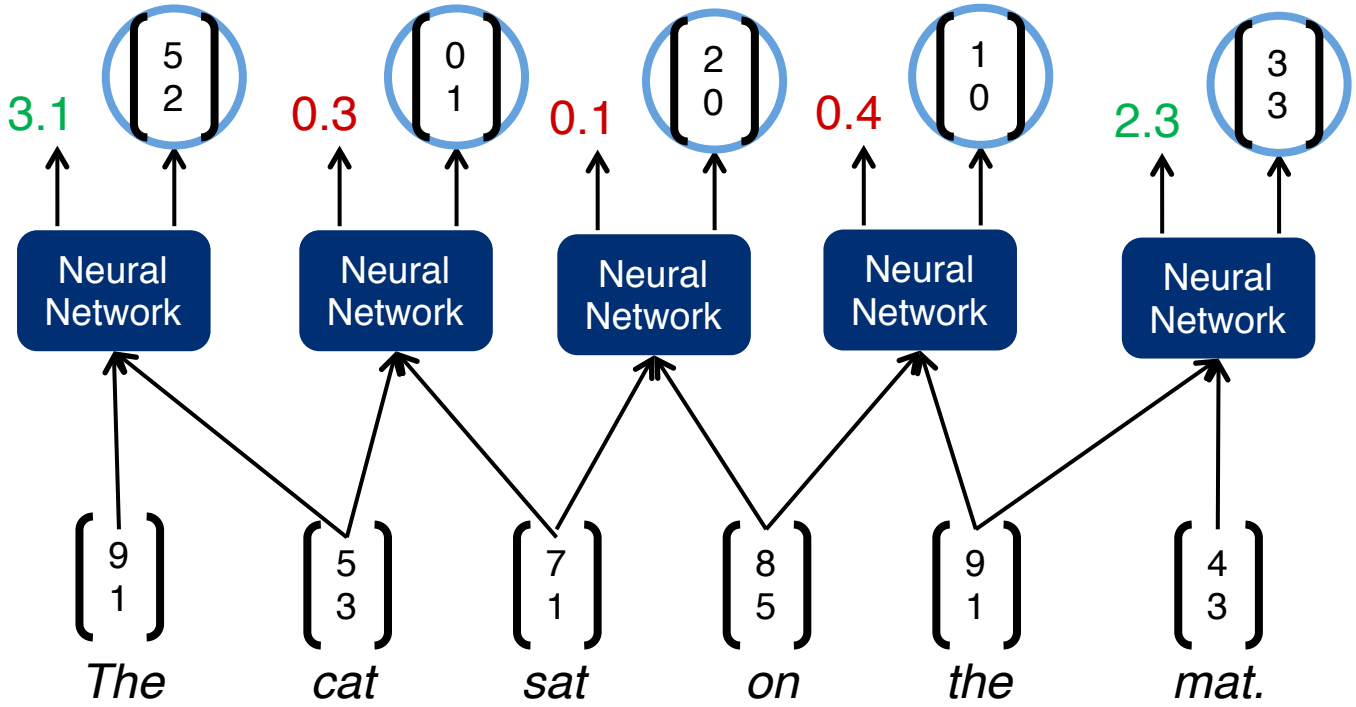
where tanh:



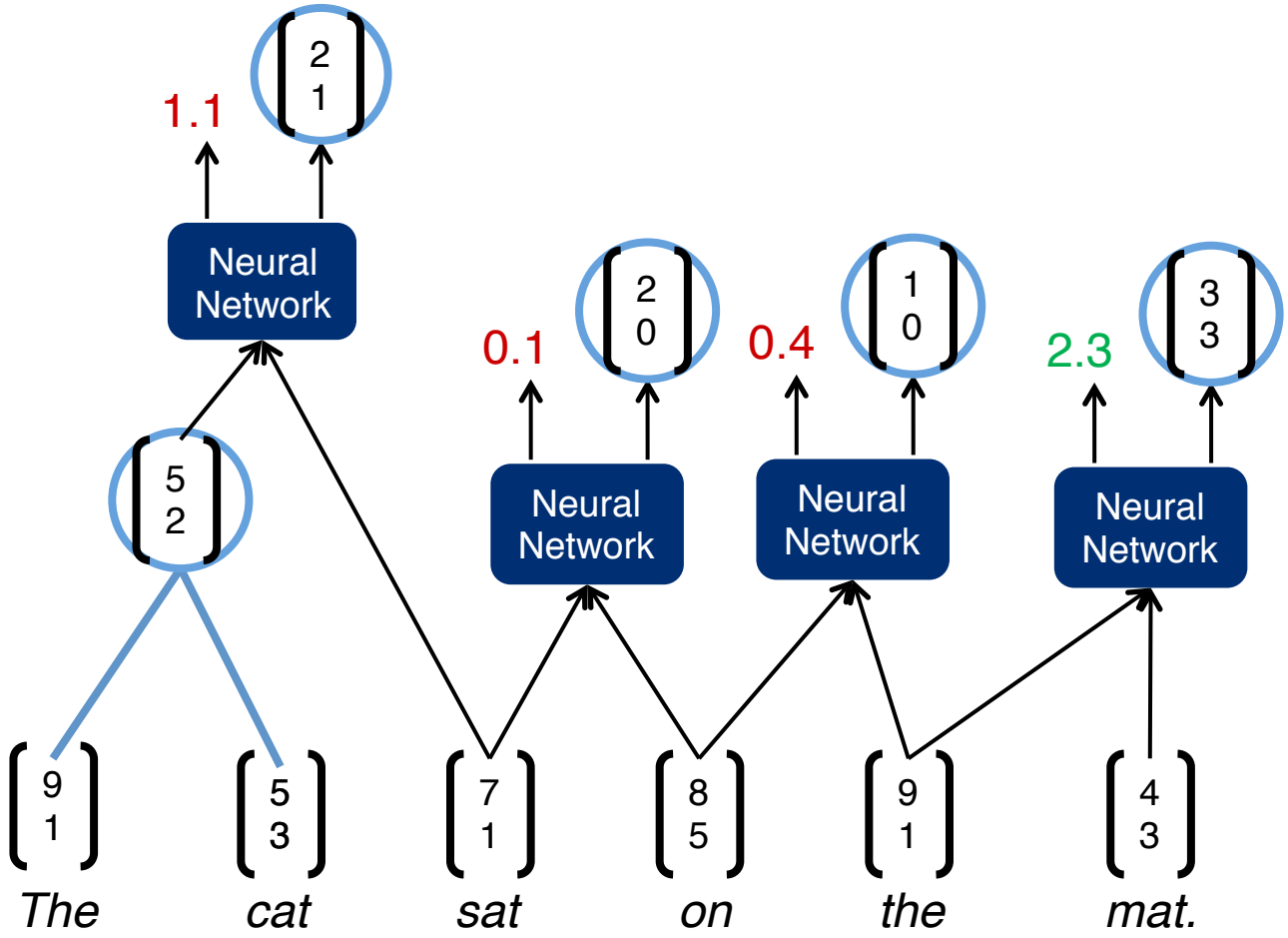
Related Recursive Neural Network based approaches

- Previous RNN work (Goller & Küchler (1996), Costa et al. (2003)) assumed fixed tree structure or used binary vectors
- Henderson (2003), Titov & Henderson (2007) use NN for parse decision prediction but not representation
- Pollack (1990): Recursive auto-associative memories (RAAMs)

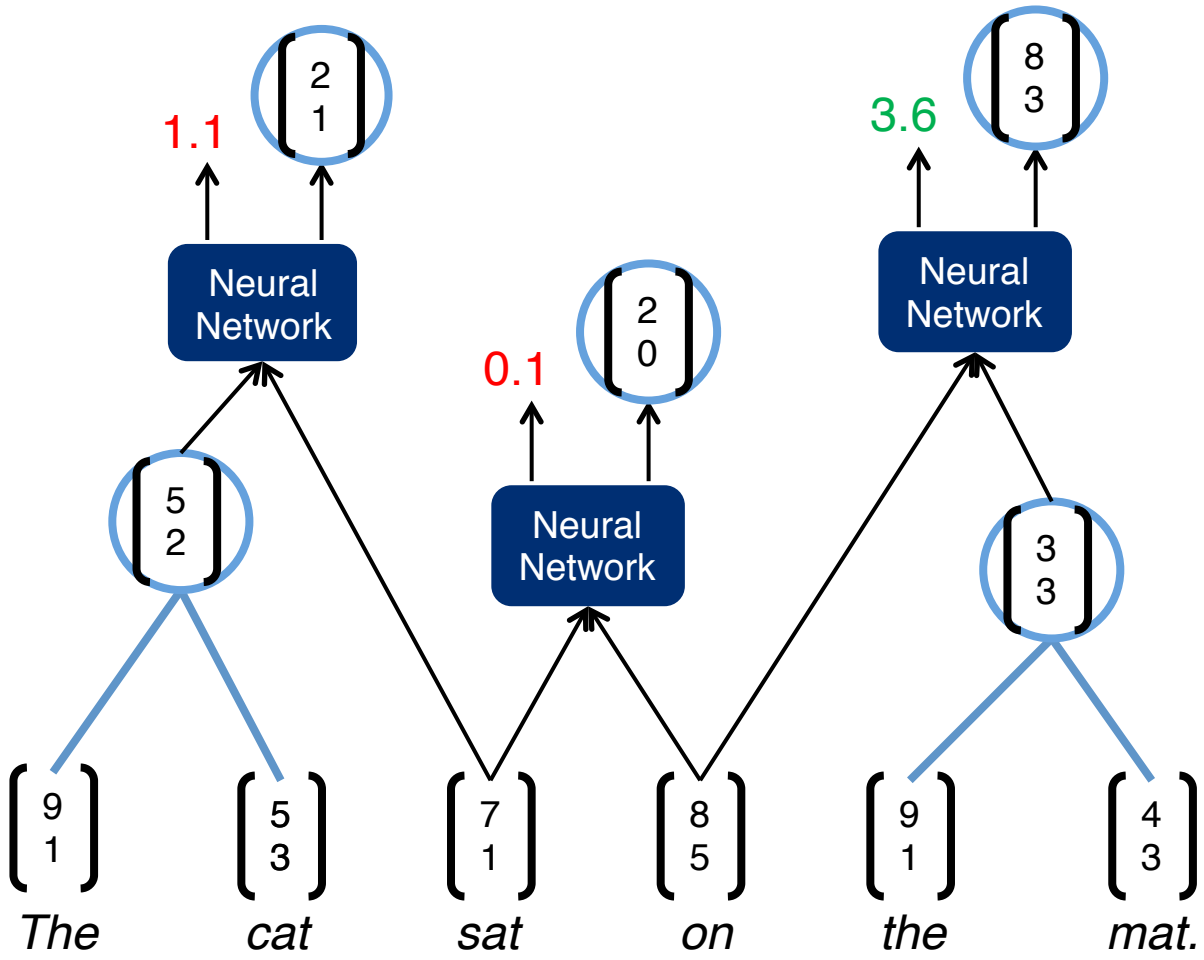
Parsing a sentence



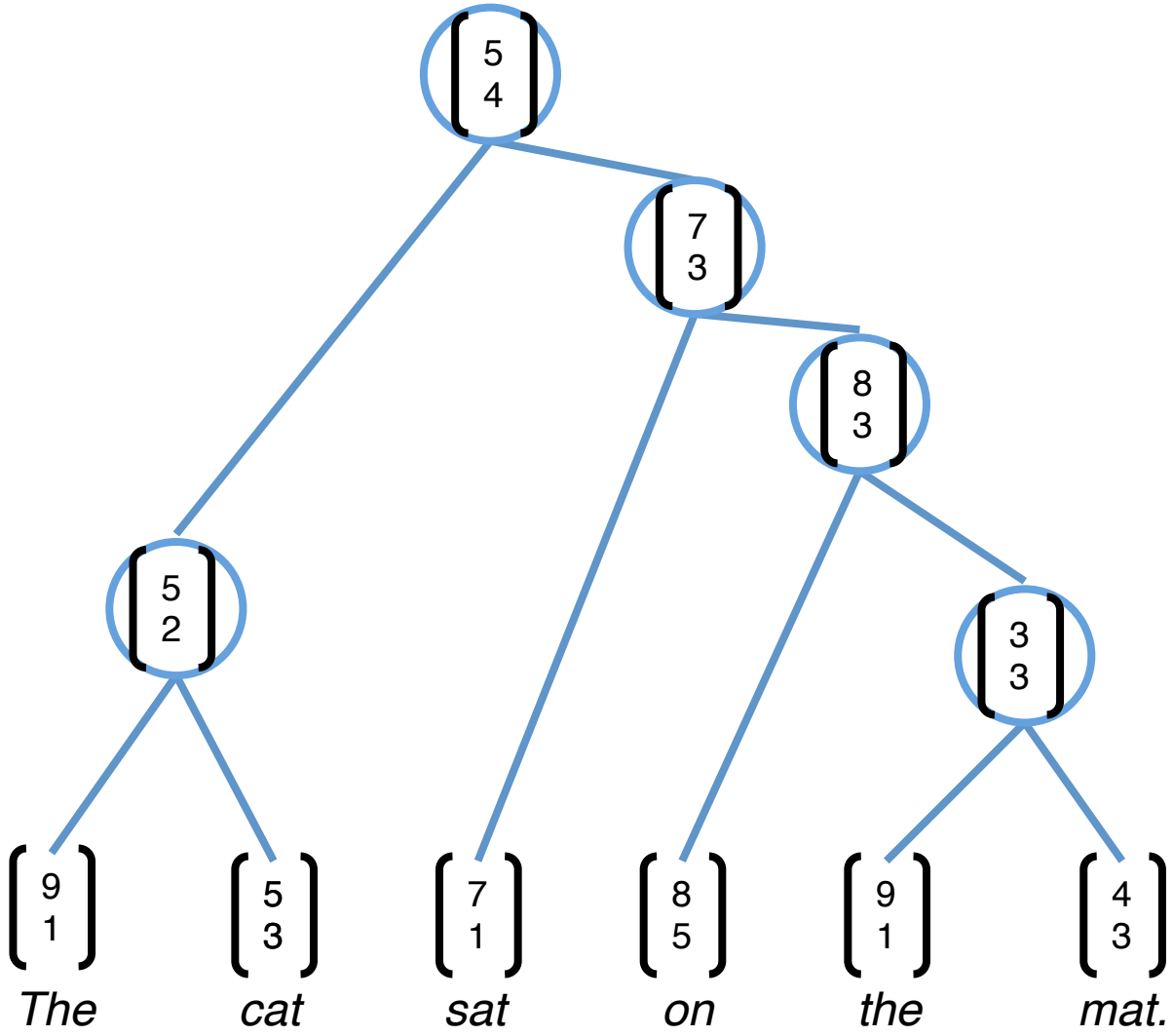
Parsing a sentence



Parsing a sentence



Parsing a sentence



Discussion: Simple RNN

State of the art in paraphrase detection (Socher et al. 2011)

Semantic similarity / nearest neighbors

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

Paraphrase detection

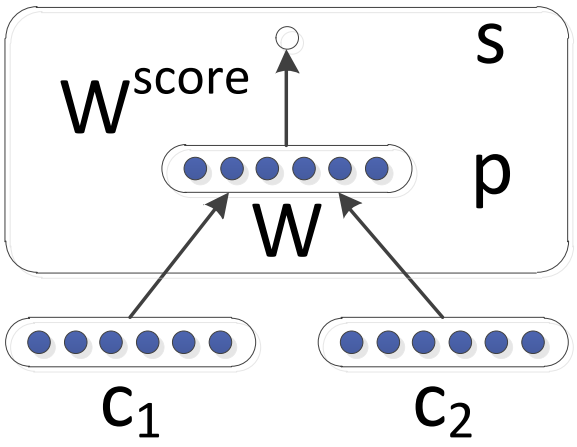
System	P	R	F1
Bannard & Callison-Burch 2005	0.28	0.12	0.17
Callison-Burch 2008 CB7	0.52	0.17	0.26
RNN (this work)	0.38	**0.64**	**0.48**
RNN-CB7 (RNN & CB7 combined)	**0.53**	0.20	0.29

Discussion: Simple RNN

Only a single weight matrix

Could capture some phenomena

Not a state-of-the-art parser



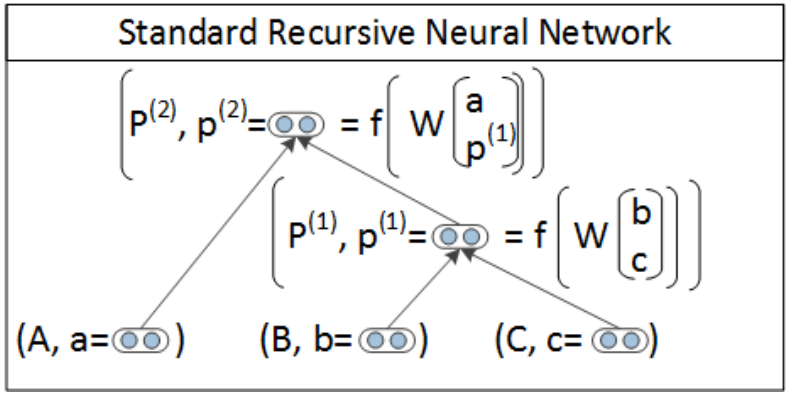
Not adequate for more complex, rich composition structure, or for parsing long sentences

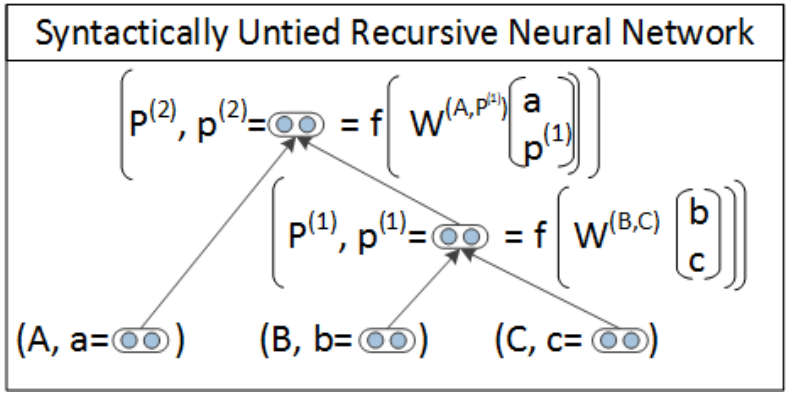
The composition function is the same for all syntactic categories, punctuation, etc.

Solution 1: PCFG + Syntactically-Untied RNN

Hypotheses:

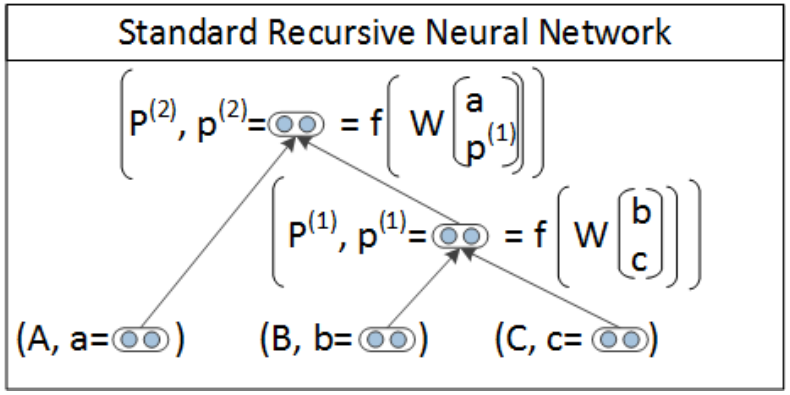
- A symbolic Context-Free Grammar (CFG) backbone is quite adequate for basic syntactic structure
- An RNN can do a fairly good job for meaning composition
- It would do better if we allow a different composition matrix for different syntactic environments

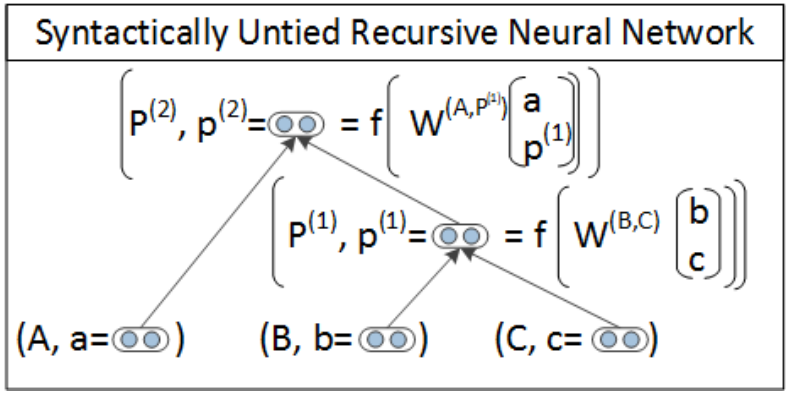




Solution 1: PCFG + Syntactically-Untied RNN

- We use the discrete syntactic categories of the children to choose the composition matrix
- Compute score using a linear combination of the RNN score + log likelihood from the PCFG rule
- PCFG backbone gives us speed by letting us prune unlikely candidates





Parsing Evaluation

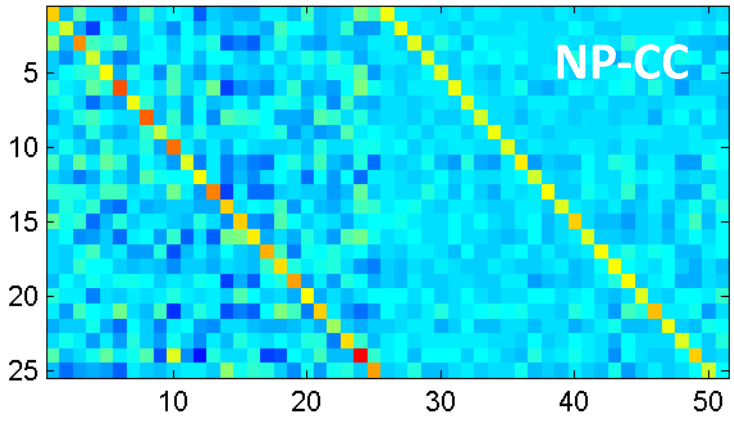
Standard *WSJ* split, labeled F_1

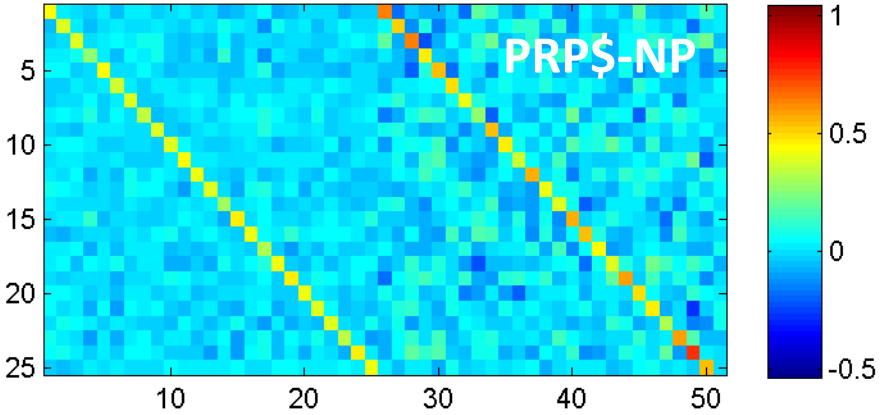
Parser	Test, All Sentences
Stanford PCFG, (Klein and Manning, 2003a)	85.5
Stanford Factored (Klein and Manning, 2003b)	86.6
Factored PCFGs (Hall and Klein, 2012)	89.4
Collins (Collins, 1997)	87.7
SSN (Henderson, 2004)	89.4
Berkeley Parser (Petrov and Klein, 2007)	90.1
CVG (RNN) (Socher et al., ACL 2013)	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

SU-RNN Analysis

Learns soft notion of head words

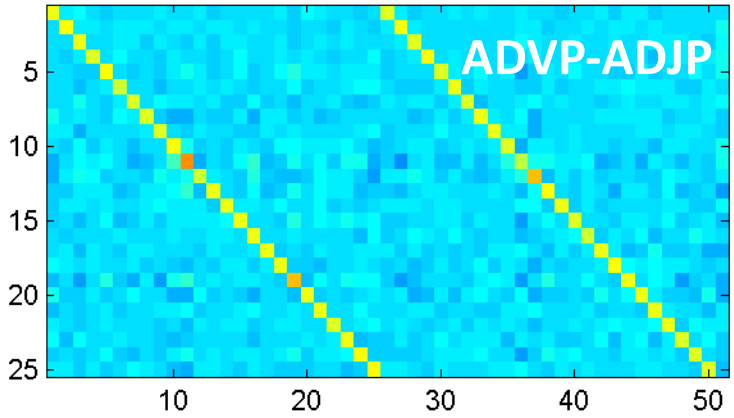
Initialization: $W^{(\cdot\cdot)} = 0.5[I_{n\times n} I_{n\times n} 0_{n\times 1}] + \epsilon$

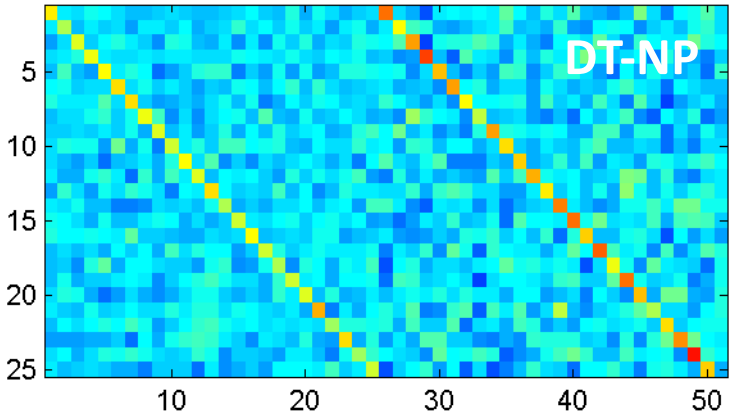


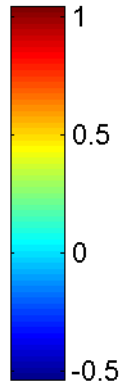


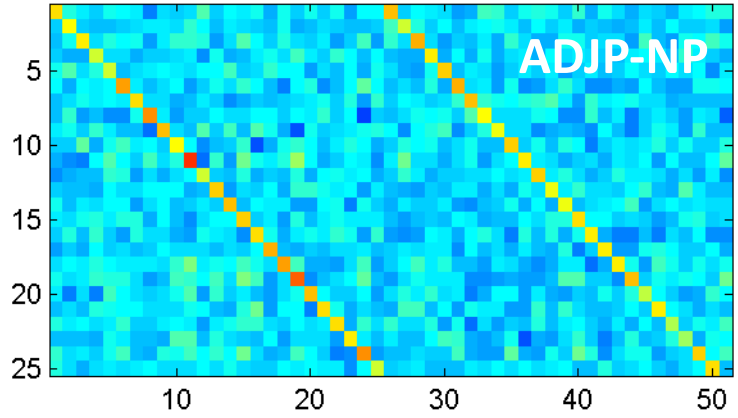
SU-RNN Analysis

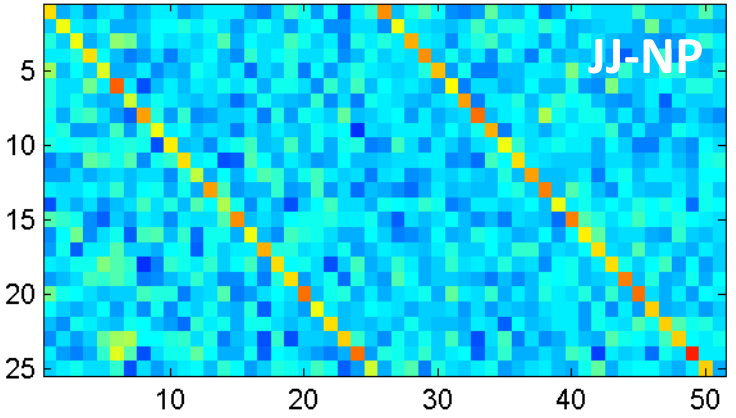
Some consistency and strength in “semantic head” modifiers

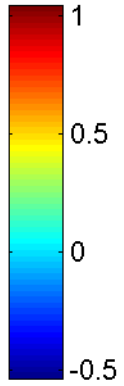






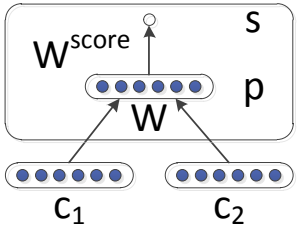






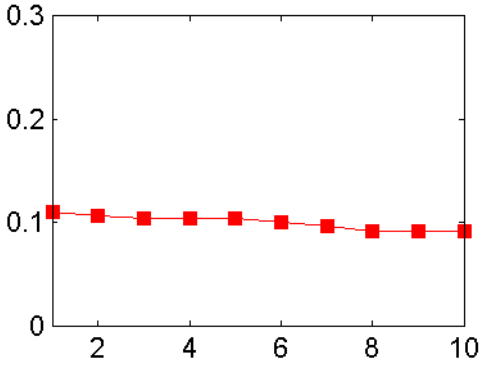
(SU-)RNNs: limitations of vector compositionality

Basic RNN composition function does not allow strong interactions between child vectors

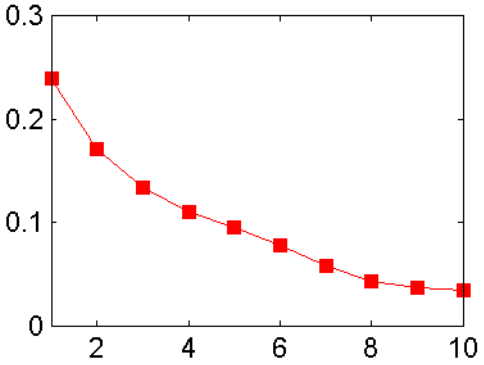


$$p = f\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

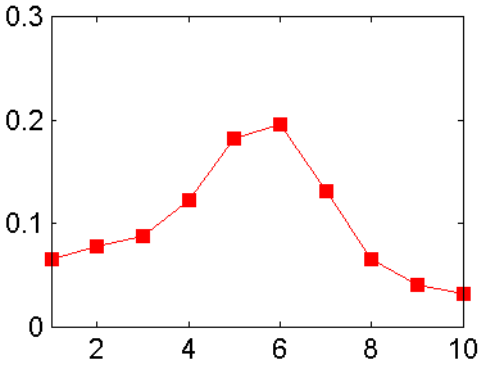
not



bad



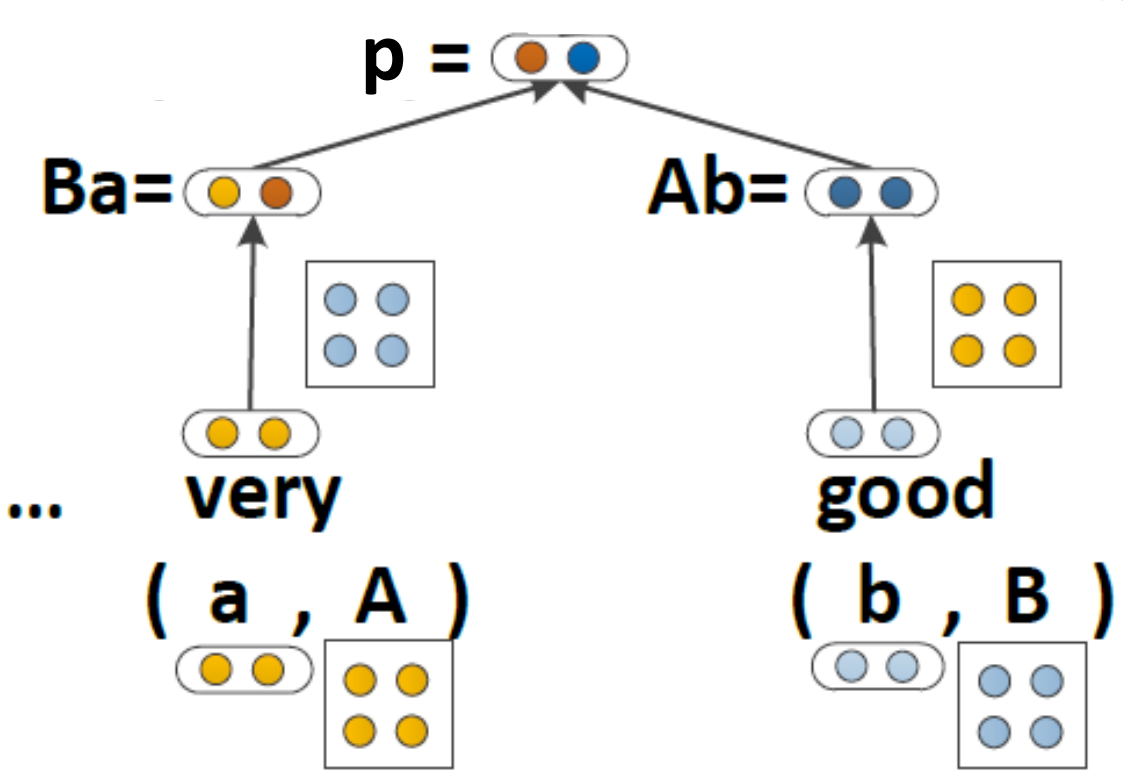
not bad



Solution 2: Matrix-vector RNNs

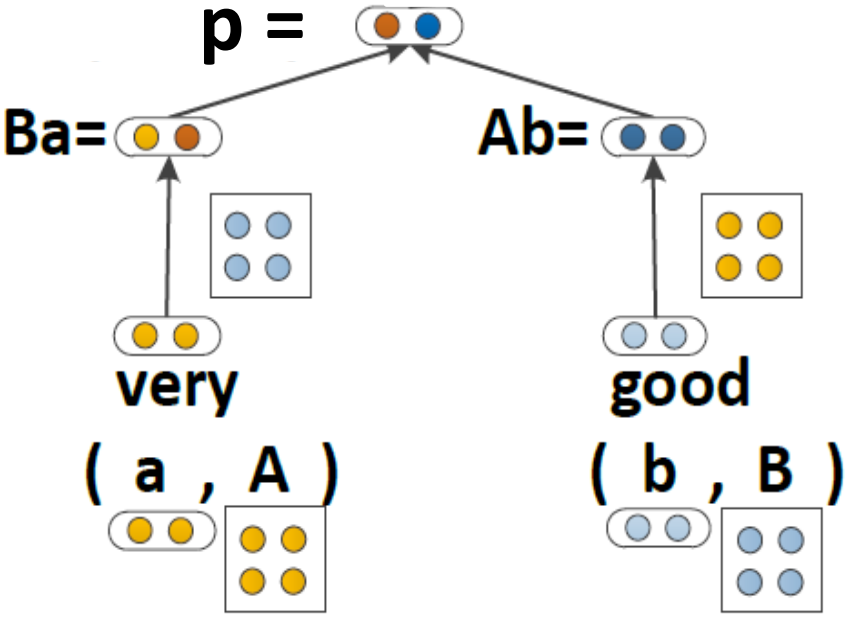
$$p = f\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

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



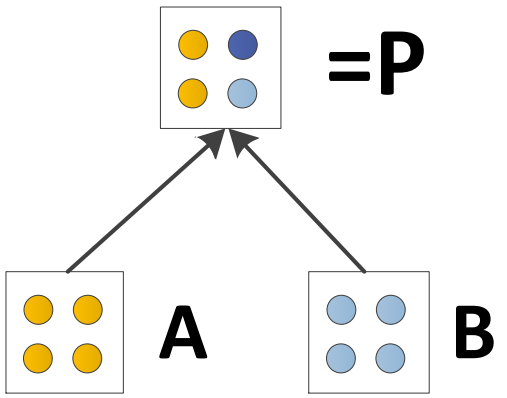
Solution 2: Matrix-vector RNNs

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



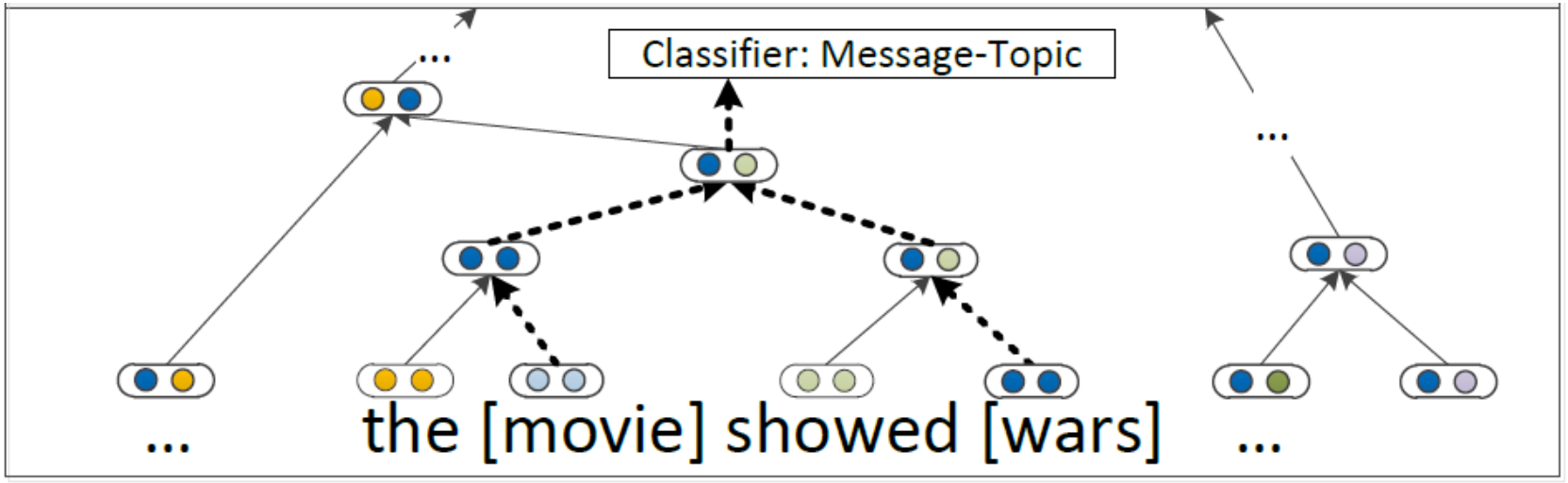
$$P = g(A, B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$$

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



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

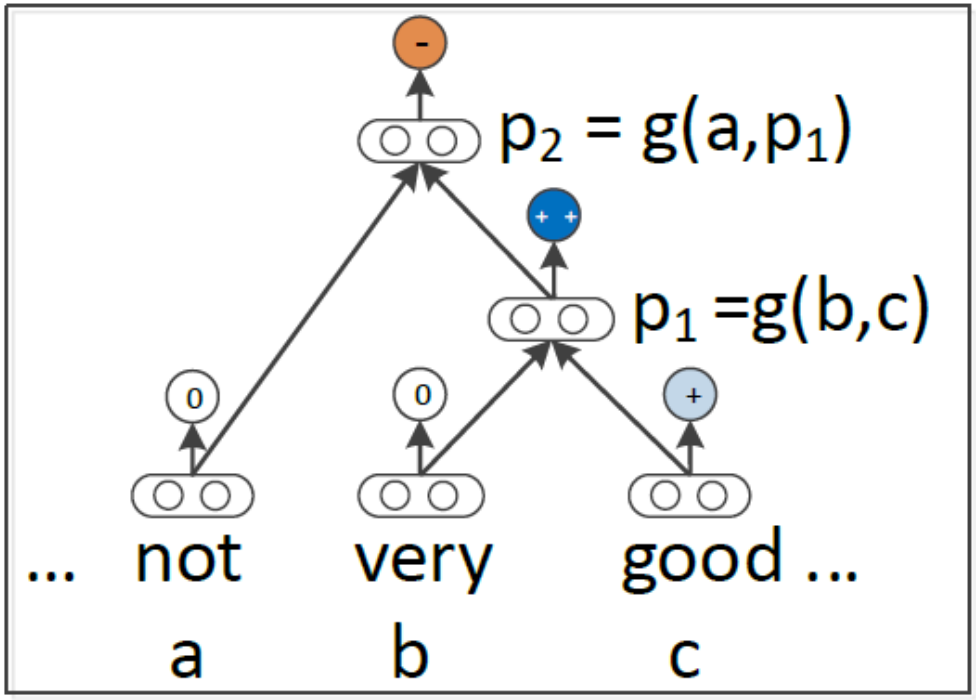


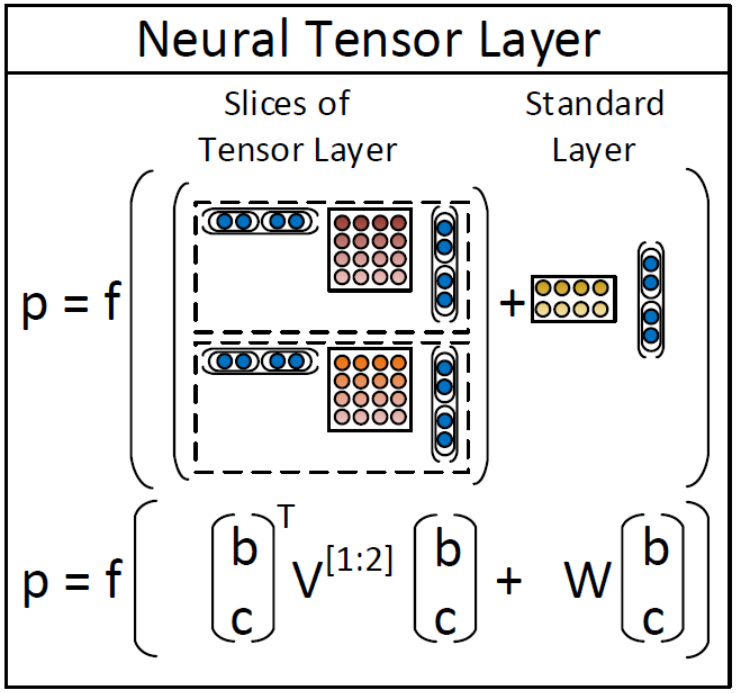
Classification of Semantic Relationships

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**

Solution 3: Recursive Neural Tensor Network

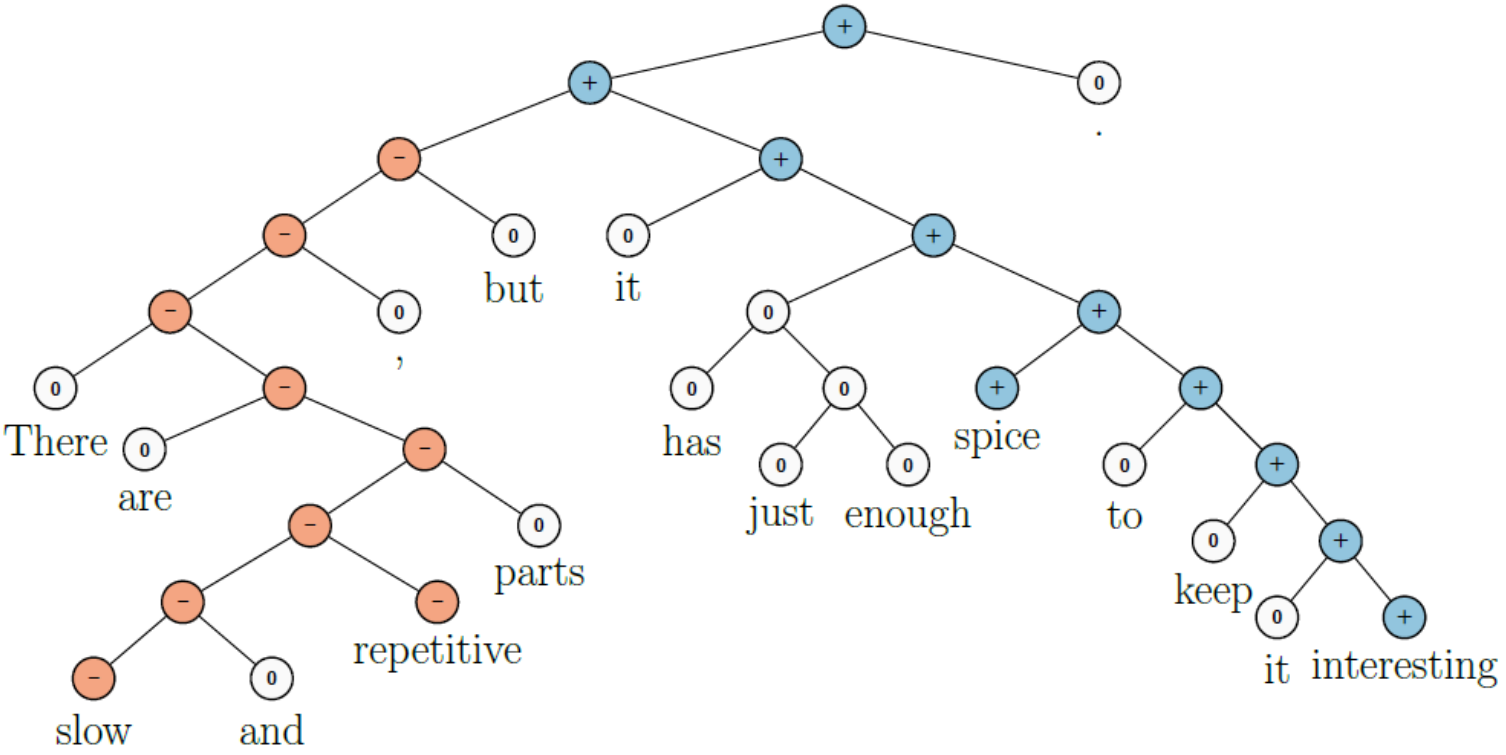
- Less parameters than MV-RNN
- Still allows the two word or phrase vectors to interact multiplicatively





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%)

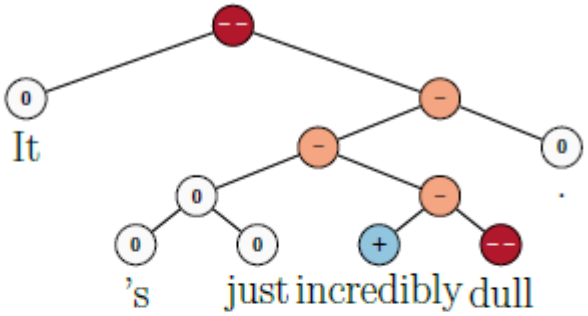


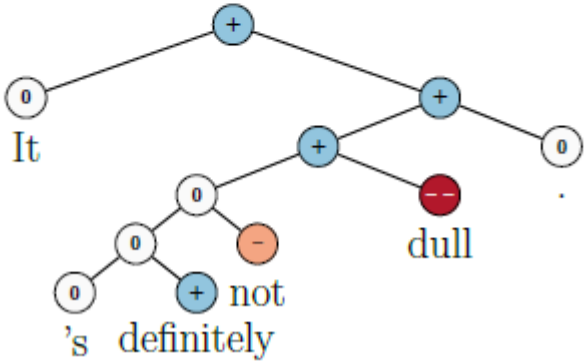
RNTN Sentiment prediction

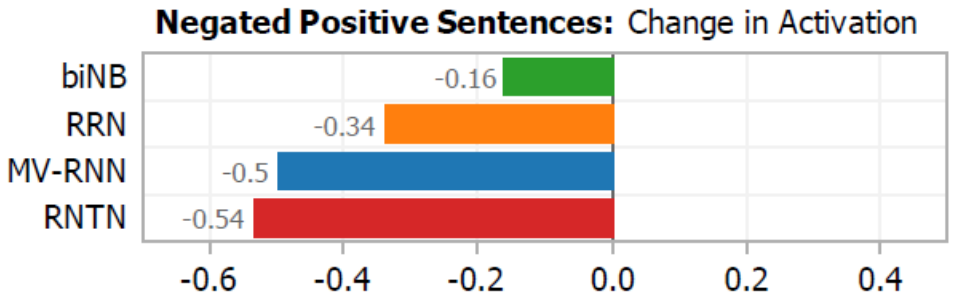
Model	Root node pos/neg
SVM	79.4
Naïve Bayes	81.8
Biword Naïve Bayes	83.1
RNN	82.4
MV-RNN	82.9
RNTN	85.4

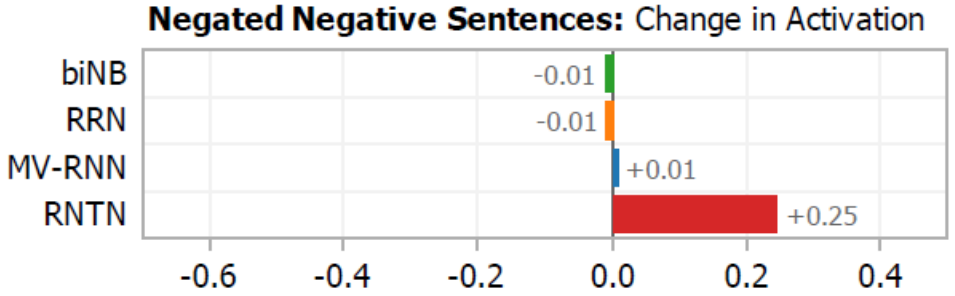
Negation Results

When negating negatives, positive activation should increase!









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

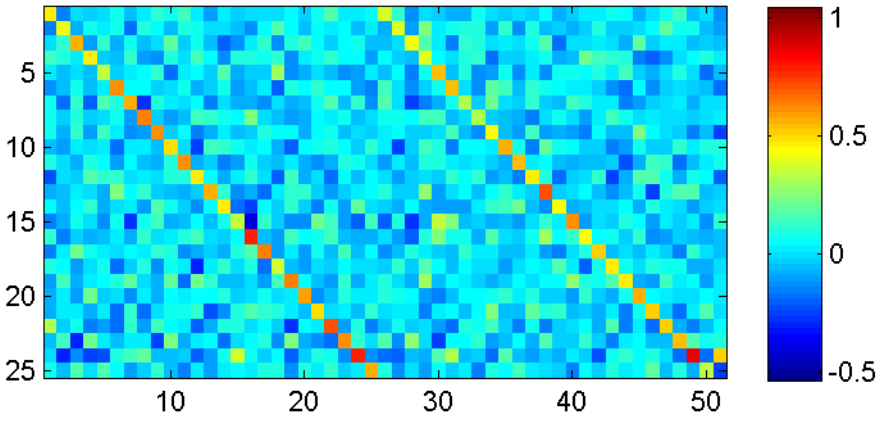
Conclusion

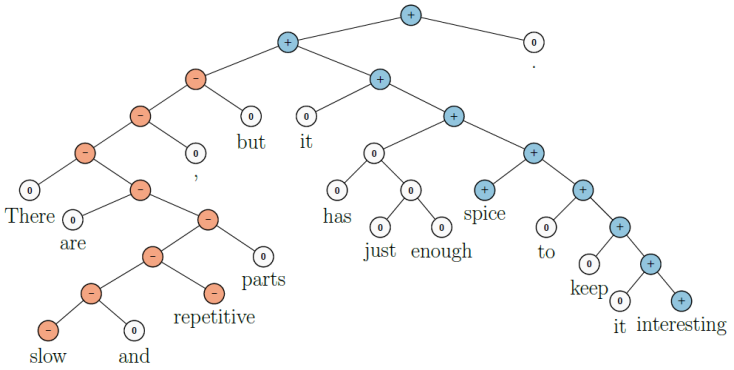
Developing intelligent machines involves being able to recognize and exploit compositional structure

It also involves other things like top-down prediction, of course

I've shown 3 successful Recursive NN techniques, for parsing, relation extraction, and sentiment analysis

Thank you!





ML Hipster

@ML_Hipster

I once trained an SVM on digits of π until it realised we're all just eigenvectors of the universe's Hamiltonian—Now THAT was deep learning.

6 Jun