CS291K – Advanced Data Mining

Instructor: Xifeng Yan
Computer Science
University of California at Santa Barbara

Recursive Neural Networks

Lecturer: Semih Yavuz
Computer Science
University of California at Santa Barbara

Source of slides

- Deep Learning for Natural Language Processing Course by Richard Socher - 2015 Stanford
- Lecture Notes by Prof. Charles Elkan: http://cseweb.ucsd.edu/~elkan/250B/learningmeaning.pdf
- Deep Learning for NLP (without Magic) by Richard Socher - NAACL 2013

Outline

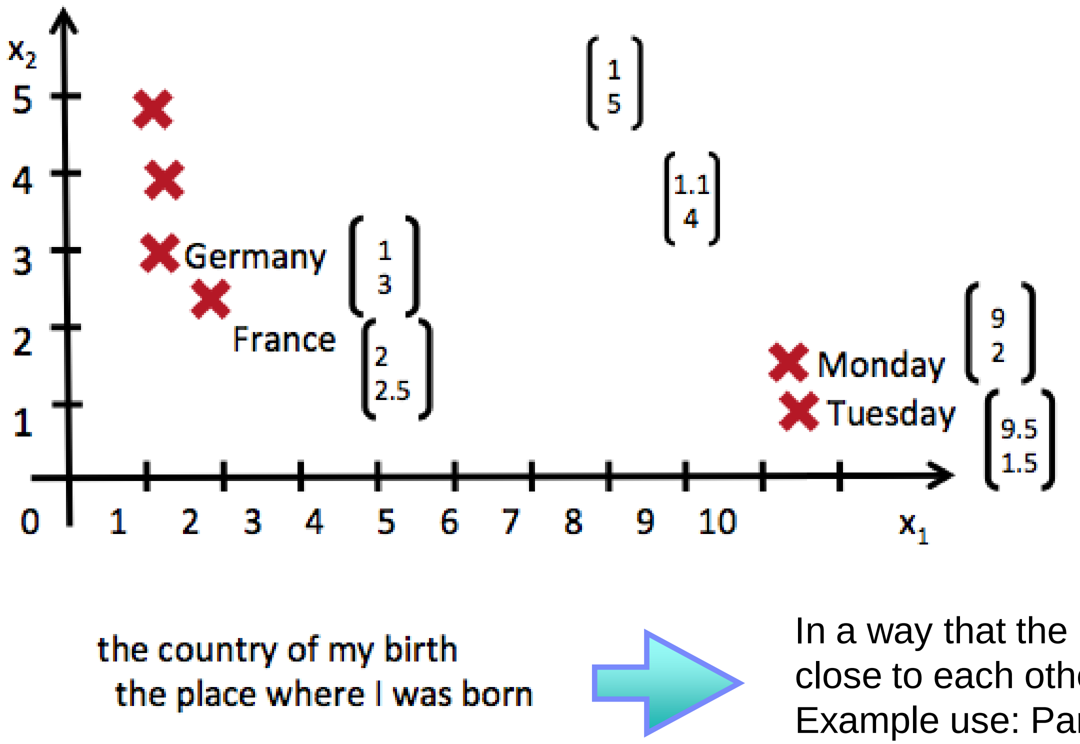
- Motivation

- Recursive Neural Networks
  - Compositionality
  - Structure Prediction: Parsing
  - Backpropagation Through Structure
  - Minor twist leads to an improved Parser

- Matrix-Vector RNNs

Word vectors

- How to represent the meaning of longer phrases?



In a way that the phrases of same meaning are close to each other.
Example use: Paraphrasing, Text Summarization

- How to map the longer phrases in to the same vector space?

Semantic Vector Space Embeddings

Many works for single word embeddings!

Ignores Word Order

Single Word Vectors

- Distributional Techniques
- Brown Clusters
- Useful as features inside models, e.g. CRFs for NER, etc.
- Cannot capture longer phrases

Documents Vectors

- Bag of words models
- PCA (LSA, LDA)
- Great for IR, document exploration, etc.
- Ignore word order, no detailed understanding

Semantic Vector Space Embeddings

Vectors representing
Phrases and Sentences
that do not ignore word order
and capture semantics for NLP tasks

Single Word Vectors

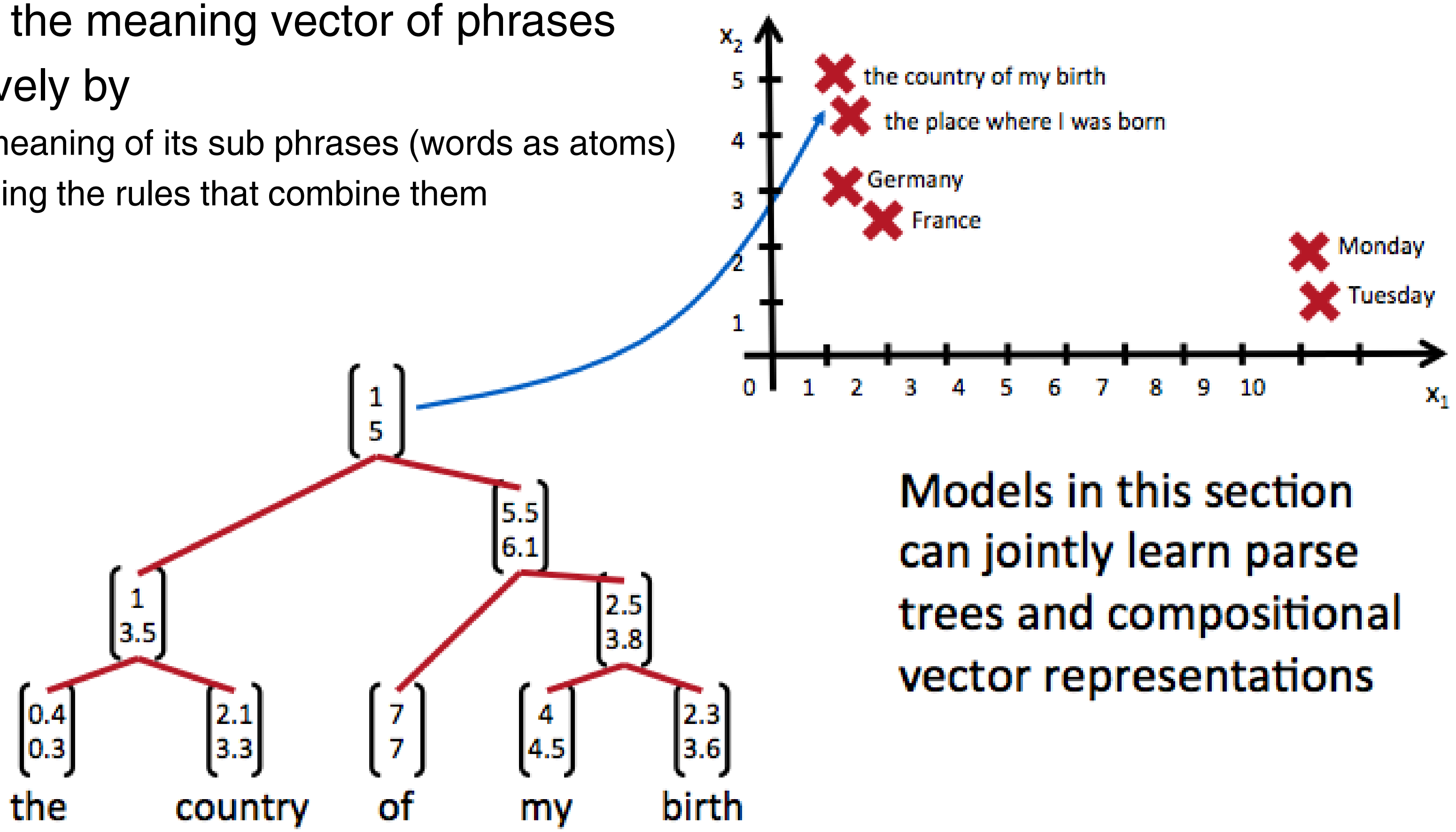
- Distributional Techniques
- Brown Clusters
- Useful as features inside models, e.g. CRFs for NER, etc.
- Cannot capture longer phrases

Documents Vectors

- Bag of words models
- PCA (LSA, LDA)
- Great for IR, document exploration, etc.
- Ignore word order, no detailed understanding

Mapping phrases into a vector space

- How to achieve compositionality
- Define the meaning vector of phrases recursively by
  - the meaning of its sub phrases (words as atoms)
  - learning the rules that combine them

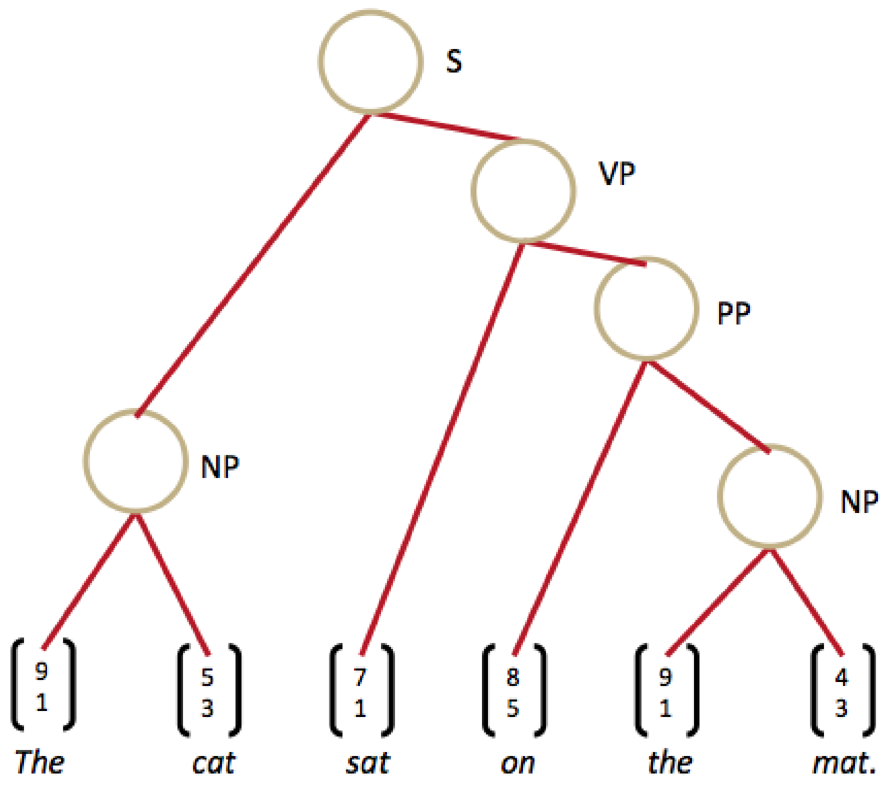


Models in this section can jointly learn parse trees and compositional vector representations

Outline

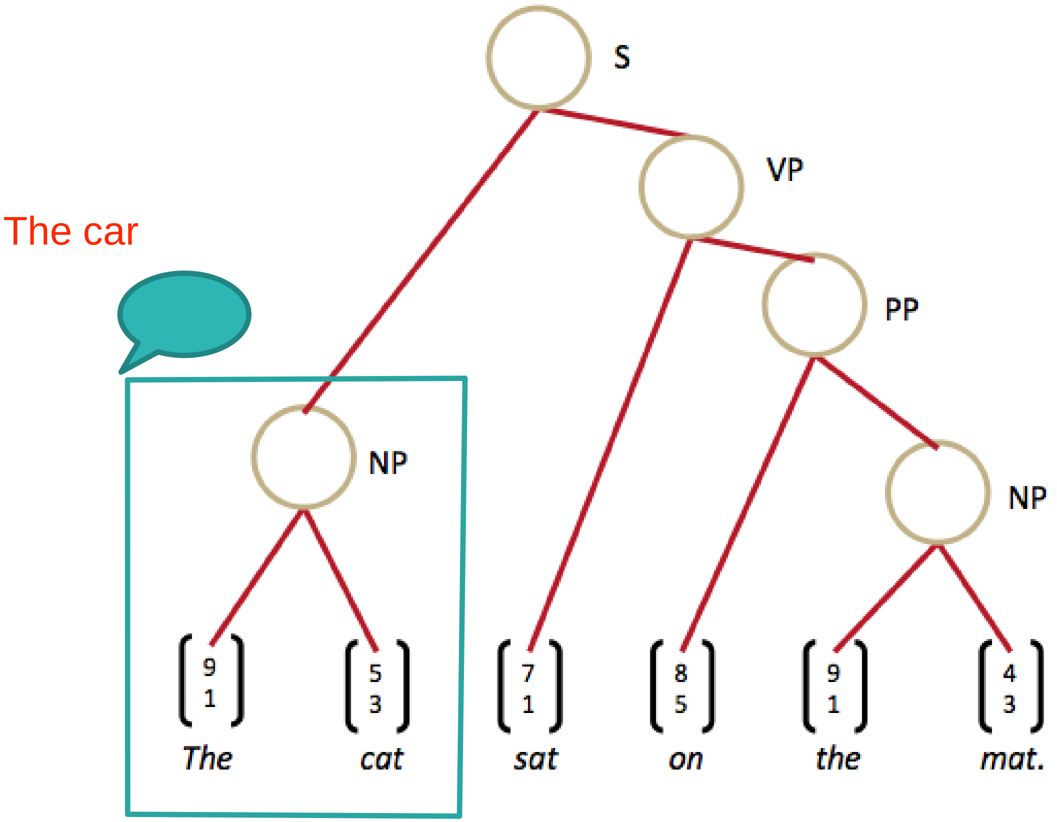
- Motivation
- Recursive Neural Networks
  - Compositionality
  - Structure Prediction: Parsing
  - Backpropagation Through Structure
  - Minor twist leads to an improved Parser
- Matrix-Vector RNNs
- Recursive Neural Tensor Networks

Parsing: What is the objective?

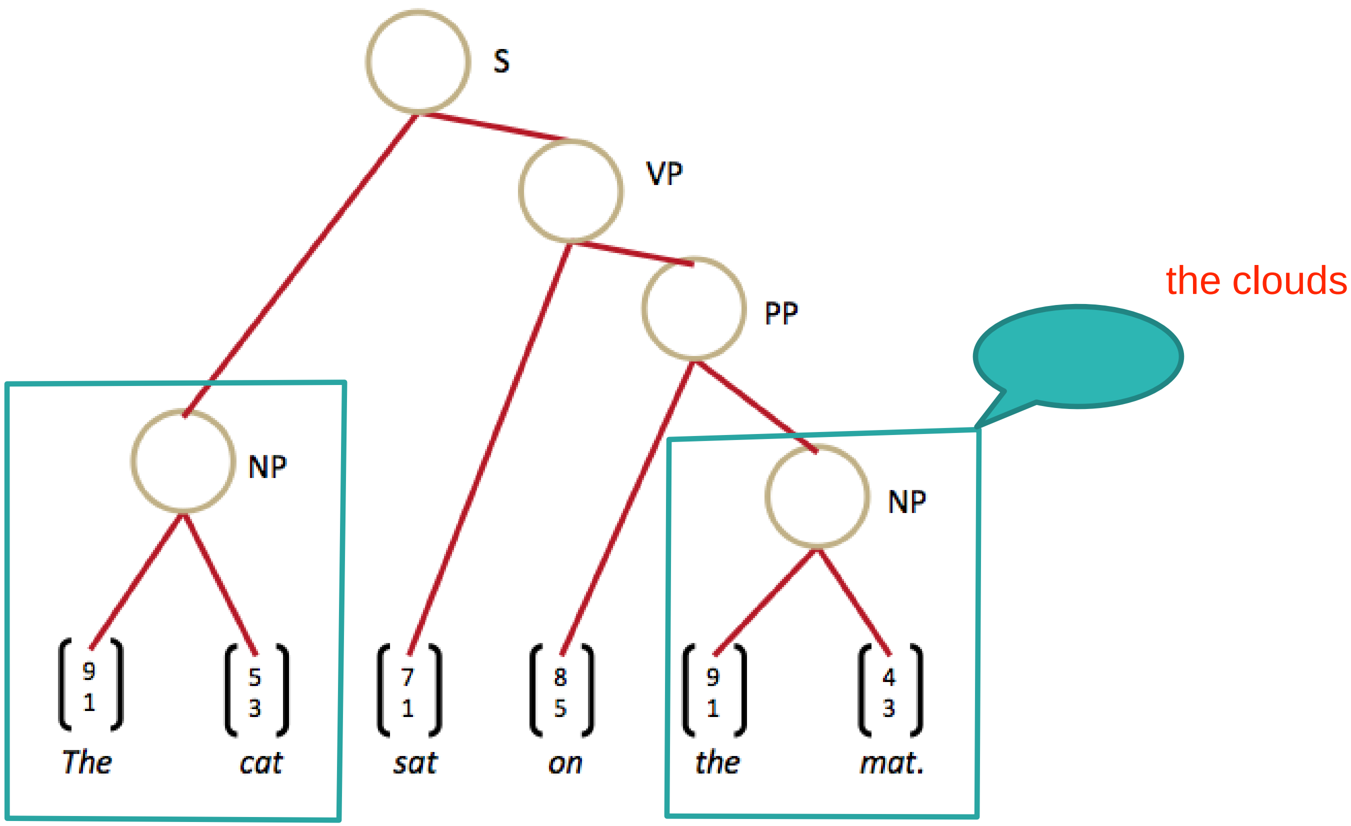


NP: Noun Phrase
VP: Verb Phrase
PP: Prepositional Phrase

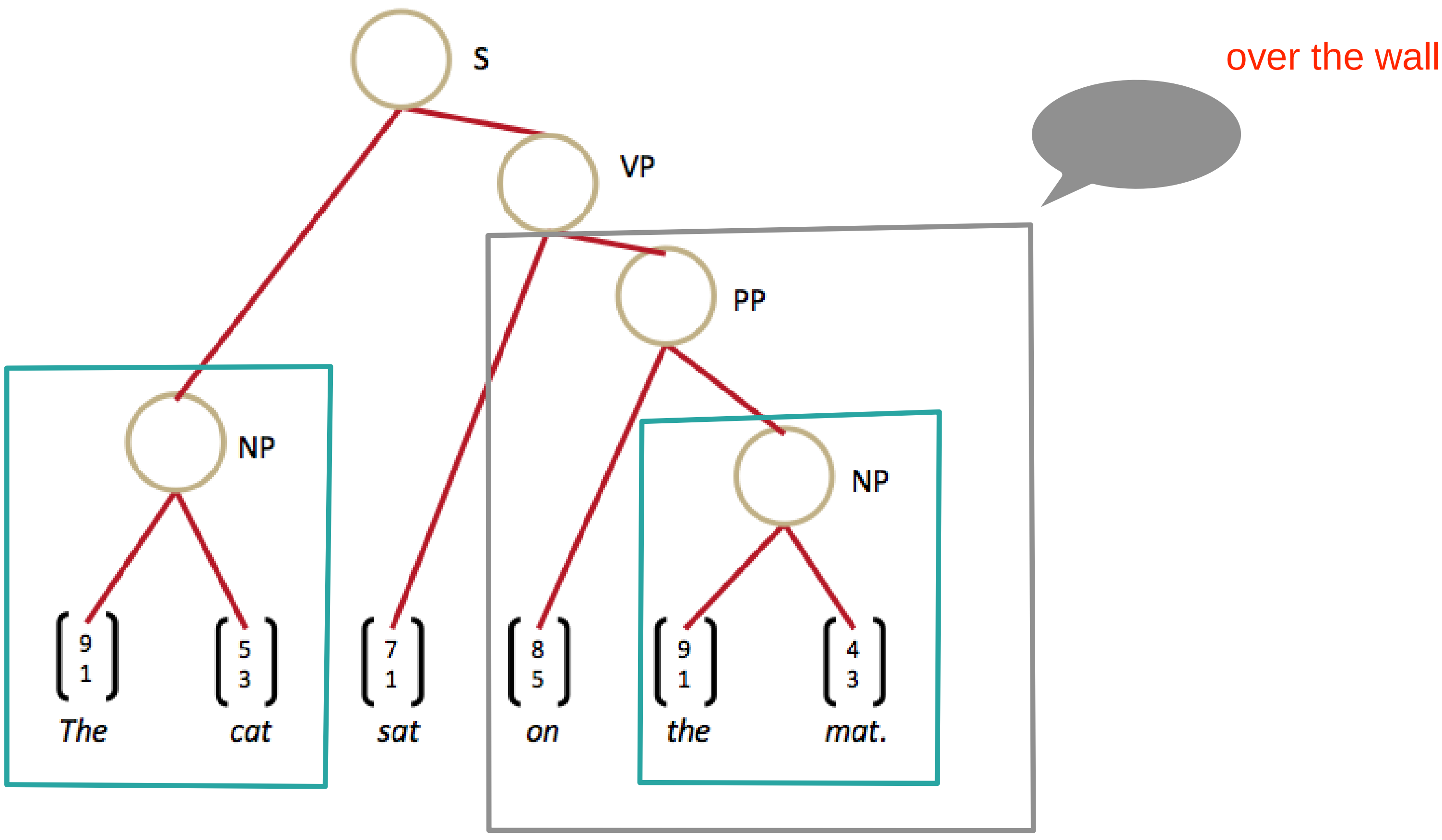
Parsing: What is the objective?



Parsing: What is the objective?

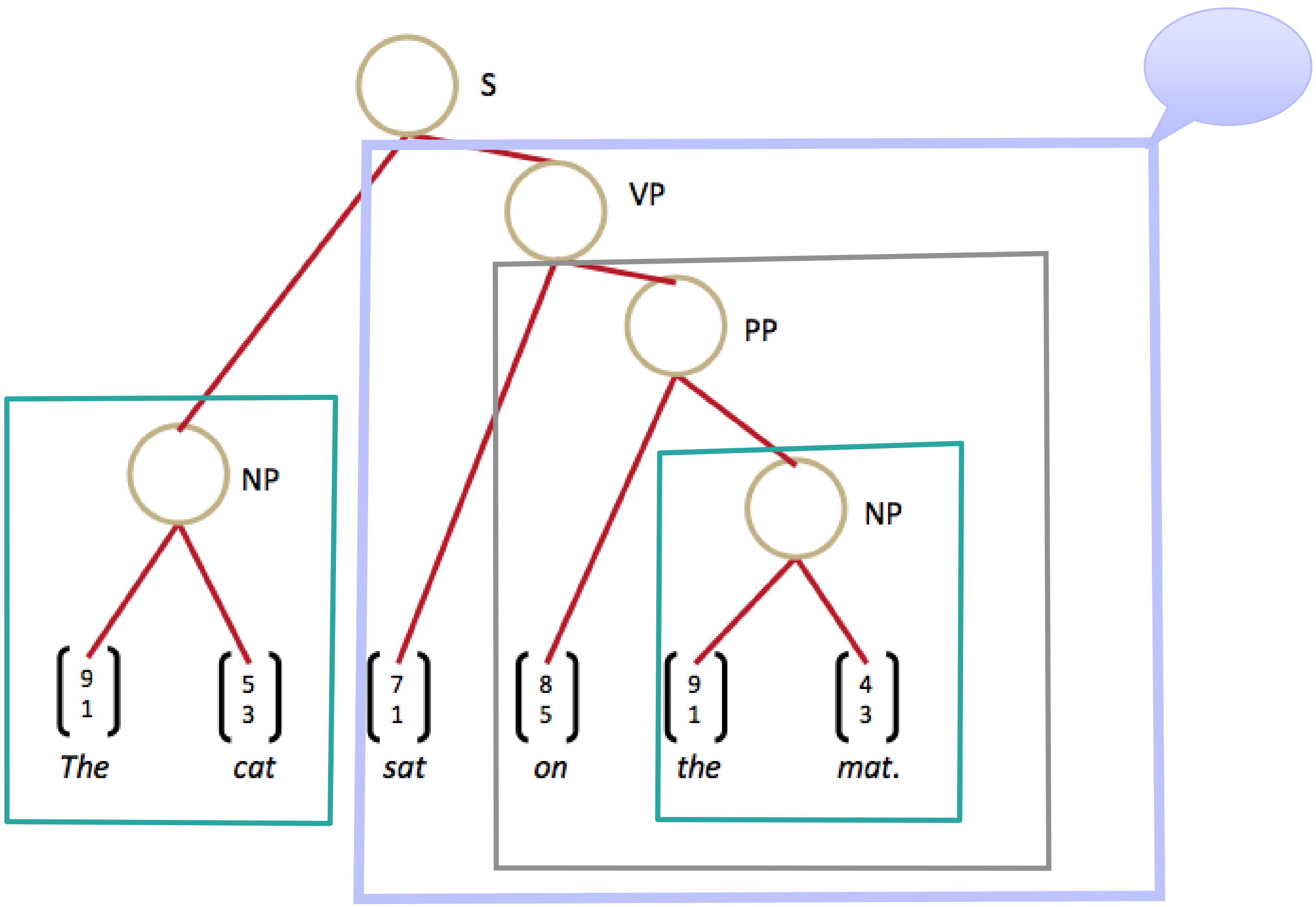


Parsing: What is the objective?

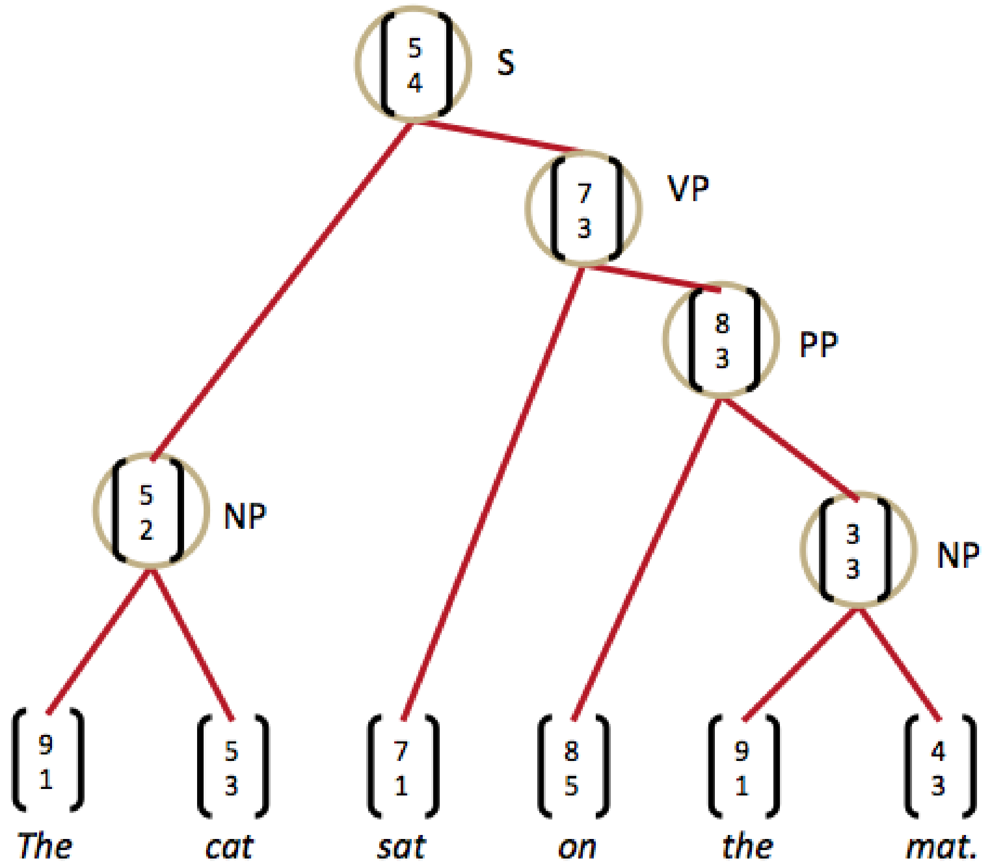


Parsing: What is the objective?

walked into the room



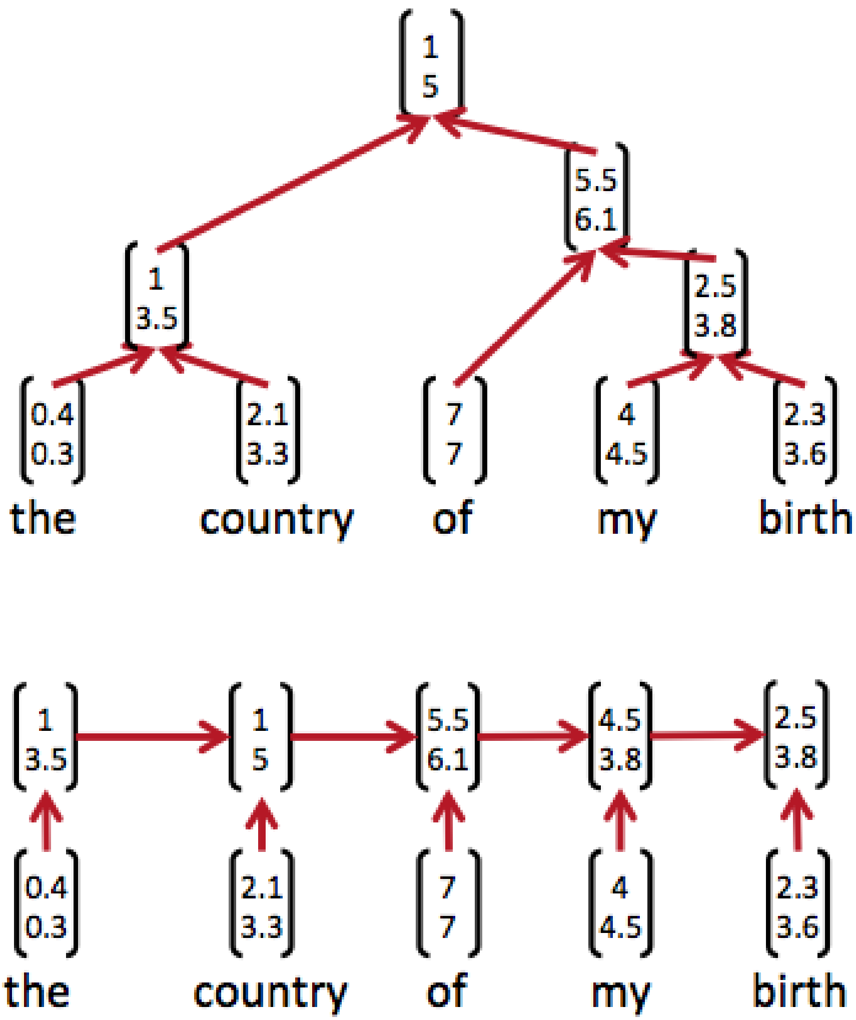
Learn jointly Structure and Representation



Learn jointly Structure and Representation

- Do we really need to learn this structure?
- Are languages structurally recursive?

Recursive vs Recurrent



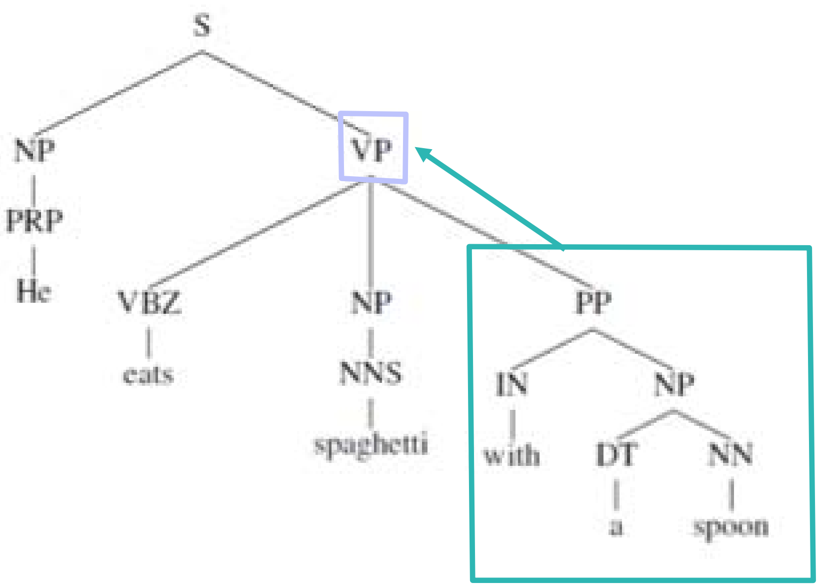
- Topology of the recurrent networks is always a chain, which is a special kind of tree.

Are languages recursive?

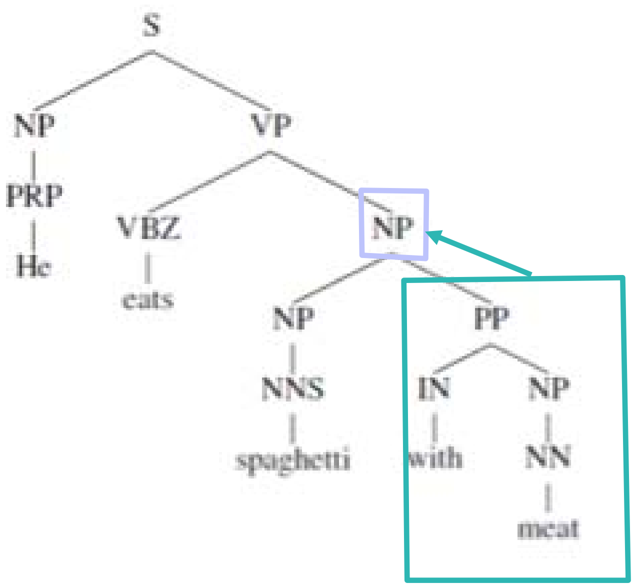
- Debatable.
- Yet, recursion is helpful in describing natural language
- Example: “the church which has nice windows”, a noun phrase containing a relative clause that contains a noun phrase. (Kind of recursive statement)

Are languages recursive?

- Argument 1: Helpful in disambiguation



PP attaches to VP



PP attaches to NP

Are languages recursive?

- Argument 2: Helpful for some tasks (e.g coreference resolution) to refer to specific phrases:
  - John and Jane went to a big festival. They enjoyed the trip and the music there.
  - “they”: John and Jane
  - “the trip”: went to a big festival
  - “there”: big festival

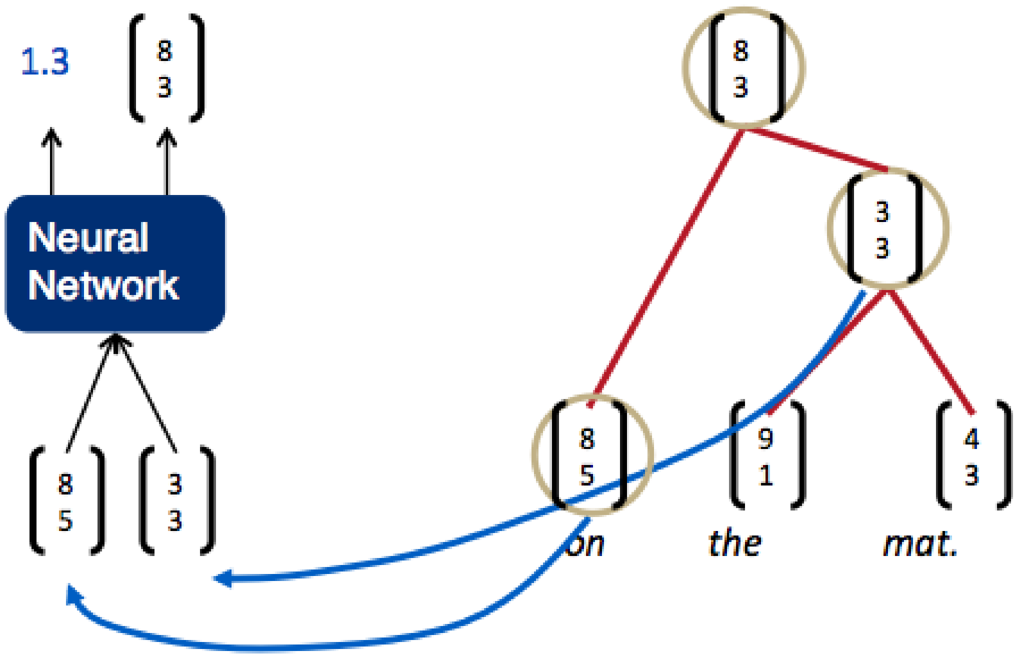
- Argument 3: Labeling becomes less clear when specific to sub phrases
  - I liked the bright screen but not the buggy slow keyboard of the phone. It was a pain to type with. It was nice to look at.

- Argument 4: Works better for some tasks to use grammatical tree structure

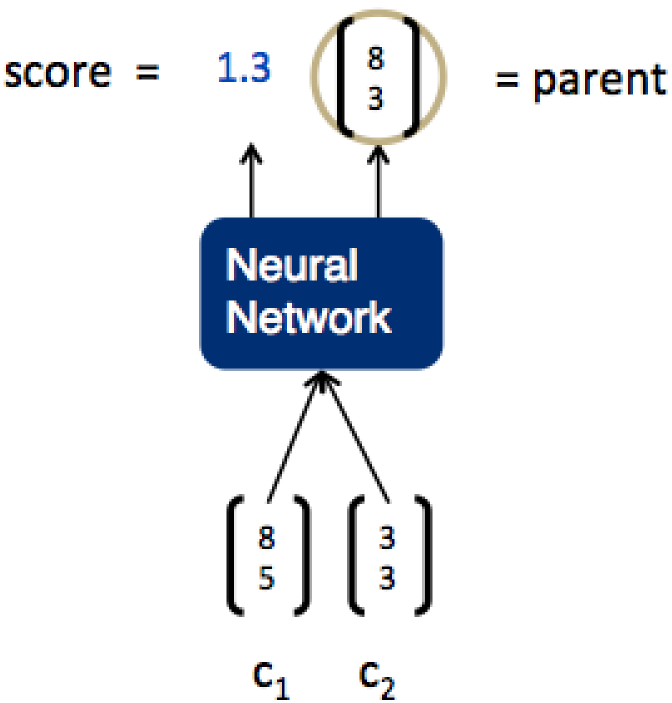
Still up for Debate!

Recursive Neural Networks for Structure Prediction

- Inputs: Two candidate children's representation
- Outputs:
    1. The semantic representation of the parent composition node
    2. Score of how plausible the new node could be



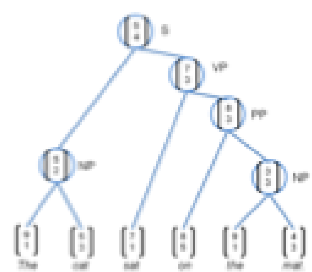
Definition of Recursive Neural Network



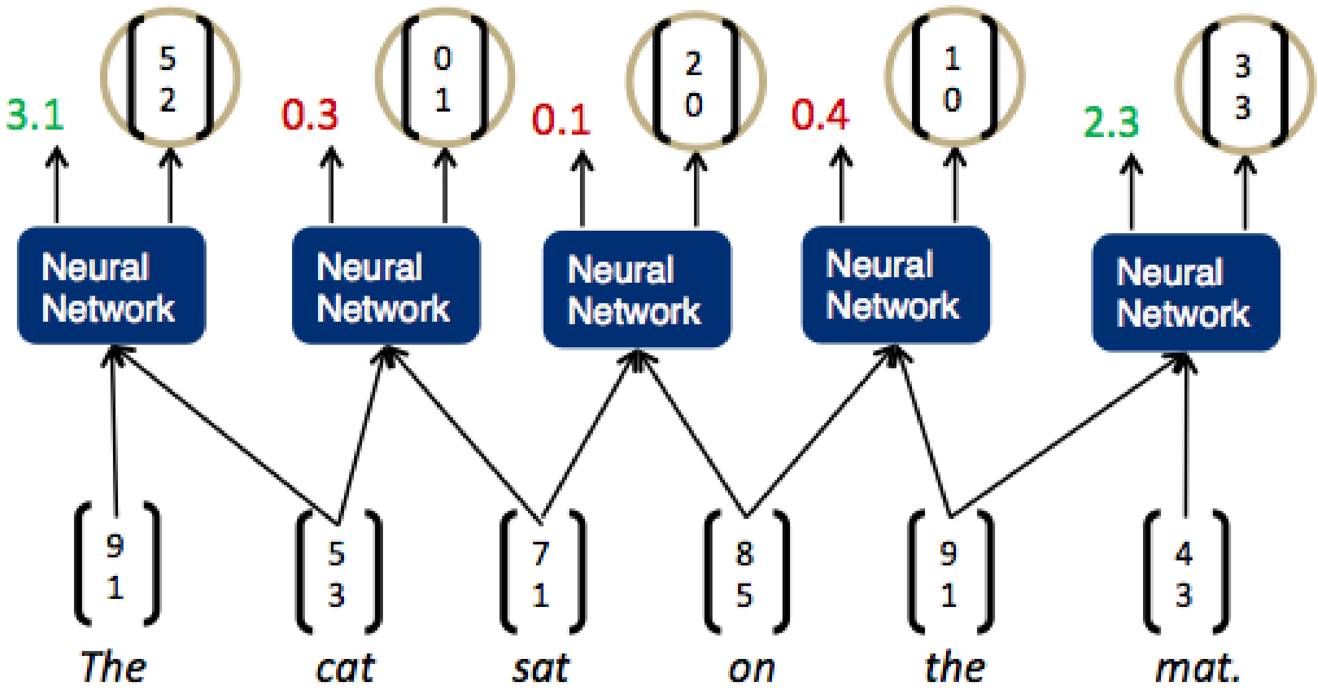
score $= U^{\mathsf{T}} p$

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

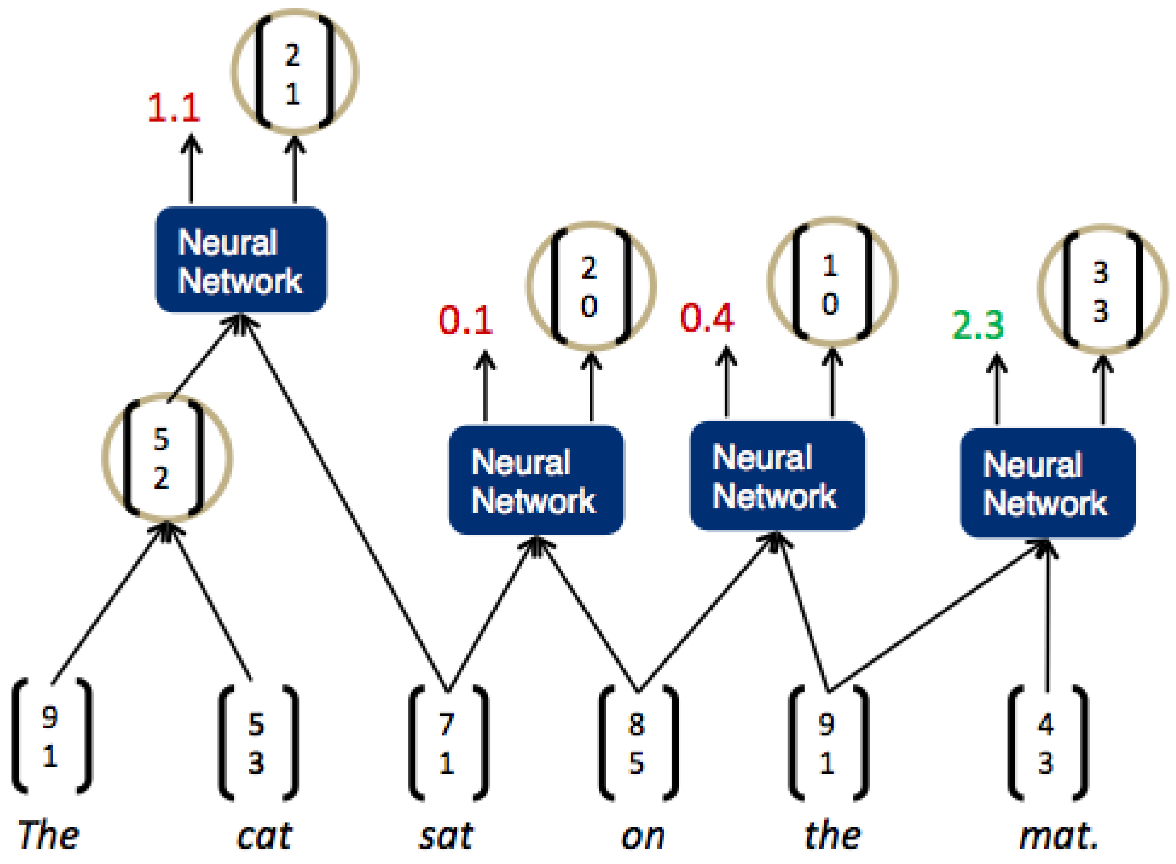
Same *W* parameters at all nodes of the tree



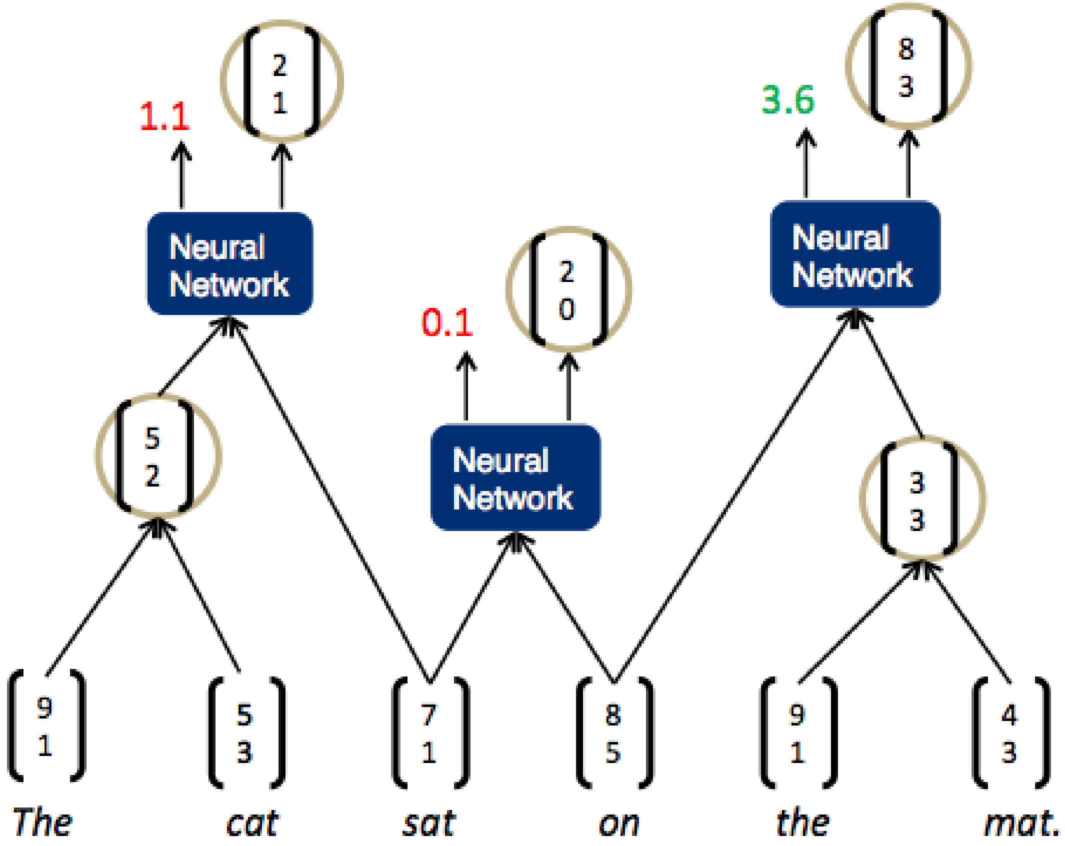
Parsing sentence with RNN



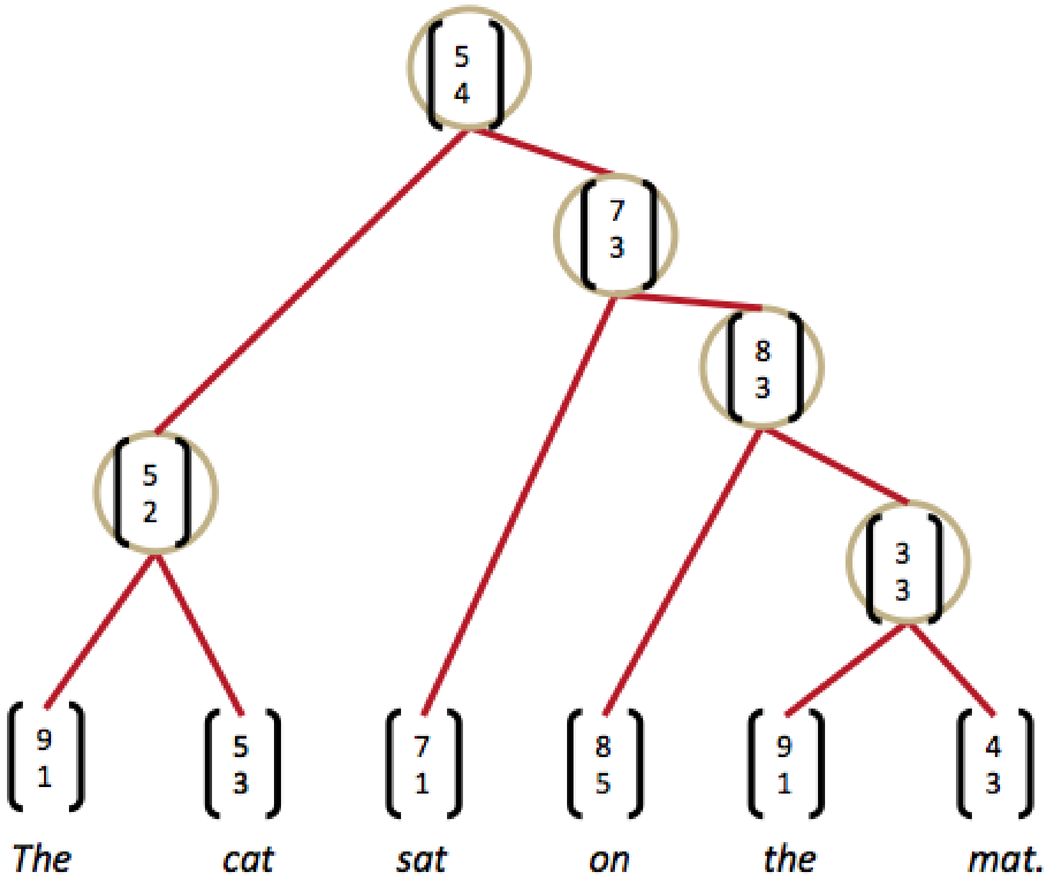
Parsing sentence with RNN



Parsing sentence with RNN



Parsing sentence with RNN



Score

- The score of a tree is computed by the sum of the parsing decision scores at each node:

$$s(x, y) = \sum_{n \in nodes(y)} s_n$$

Max-Margin Objective

- Similar to max-margin parsing (Taskar et al. 2004), a supervised max-margin objective

$$J = \sum_i s(x_i, y_i) - \max_{y \in A(x_i)} \left(s(x_i, y) + \Delta(y, y_i) \right)$$

- The loss $\Delta(y, y_i)$ penalizes all incorrect decisions

- Structure search for A(x) was maximally greedy
  - Instead: Beam Search with Chart

Backpropagation Through Structure

Introduced by Goller & Küchler (1996)

Principally the same as general backpropagation

$$\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \circ f'(z^{(l)}),$$

$$\frac{\partial}{\partial W^{(l)}} E_R = \delta^{(l+1)} (a^{(l)})^T + \lambda W^{(l)}$$

Three differences resulting from the recursion and tree structure:

1. Sum derivatives of *W* from all nodes
2. Split derivatives at each node
3. Add error messages

Backpropagation Through Structure

- Sum derivatives of all nodes

You can actually assume it's a different *W* at each node

Intuition via example:

$$
\begin{aligned}
& \frac{\partial}{\partial W} f(W(f(Wx)) \\
= \quad & f'(W(f(Wx)) \left(\left(\frac{\partial}{\partial W} W \right) f(Wx) + W \frac{\partial}{\partial W} f(Wx) \right) \\
= \quad & f'(W(f(Wx)) \left(f(Wx) + W f'(Wx)x \right)
\end{aligned}
$$

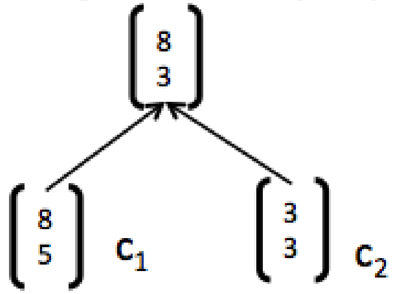
If we take separate derivatives of each occurrence, we get same:

$$
\begin{aligned}
& \frac{\partial}{\partial W_2} f(W_2(f(W_1x)) + \frac{\partial}{\partial W_1} f(W_2(f(W_1x)) \\
= \quad & f'(W_2(f(W_1x)) \left(f(W_1x) \right) + f'(W_2(f(W_1x)) \left(W_2 f'(W_1x)x \right) \\
= \quad & f'(W_2(f(W_1x)) \left(f(W_1x) + W_2 f'(W_1x)x \right) \\
= \quad & f'(W(f(Wx)) \left(f(Wx) + W f'(Wx)x \right)
\end{aligned}
$$

Backpropagation Through Structure

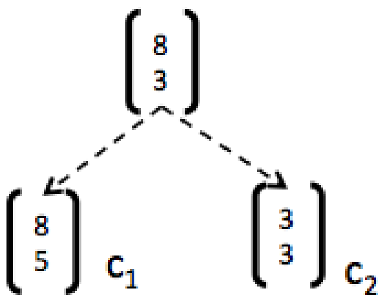
- Split derivative at each node

During forward prop, the parent is computed using 2 children



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

Hence, the errors need to be computed wrt each of them:

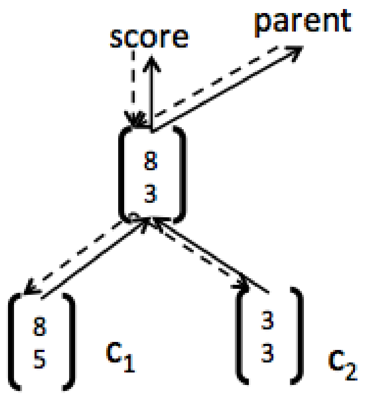


where each child's error is *n*-dimensional

$$\delta_{p\to c_1 c_2} = [\delta_{p\to c_1} \delta_{p\to c_2}]$$

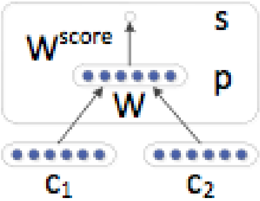
Backpropagation Through Structure

- Add error messages:
- At each node:
  - What came up (fprop) must come down (bprop)
  - Total error messages δ = error messages from parent + error message from own score



Too Simple?

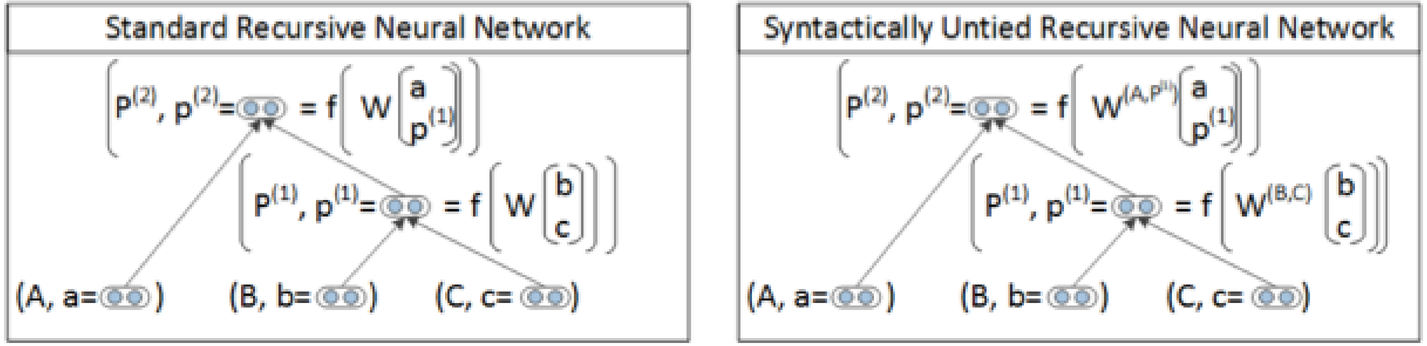
- Good results with single matrix RNN (more later)
- Single weight matrix RNN could capture some phenomena but not adequate for more complex, higher order composition and parsing long sentences



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

SU-RNN: Syntactically Untied RNN

- Idea: Condition the composition function on the syntactic categories, “untie the weights”
- Allows for different composition functions for pairs of syntactic categories, e.g. Adv + AdjP, VP + NP
- Combines discrete syntactic categories with continuous semantic information



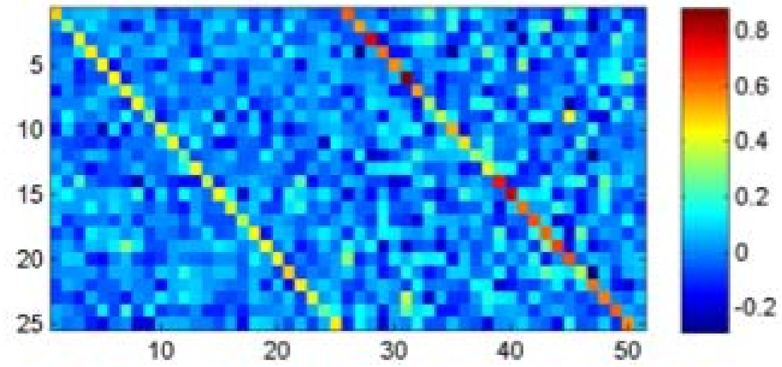
Experiments

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

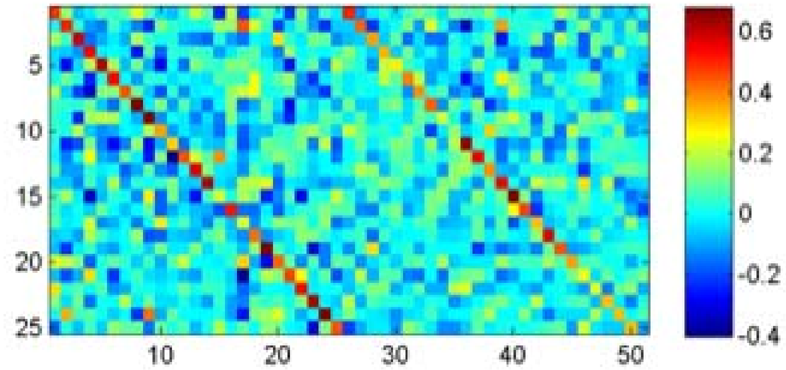
- 3.8% higher F1
- 20% faster than Stanford Factored parser

SU-RNN Analysis

- Learns notion of soft head words



DT-NP
the cat



VP-NP
read book

Analysis of resulting phrase vectors

All the figures are adjusted for seasonal variations

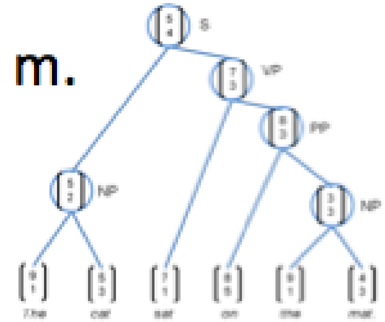
1. All the numbers are adjusted for seasonal fluctuations
2. All the figures are adjusted to remove usual seasonal patterns

Knight-Ridder wouldn't comment on the offer

1. Harsco declined to say what country placed the order
2. Coastal wouldn't disclose the terms

Sales grew almost 7% to $UNK m. from $UNK m.

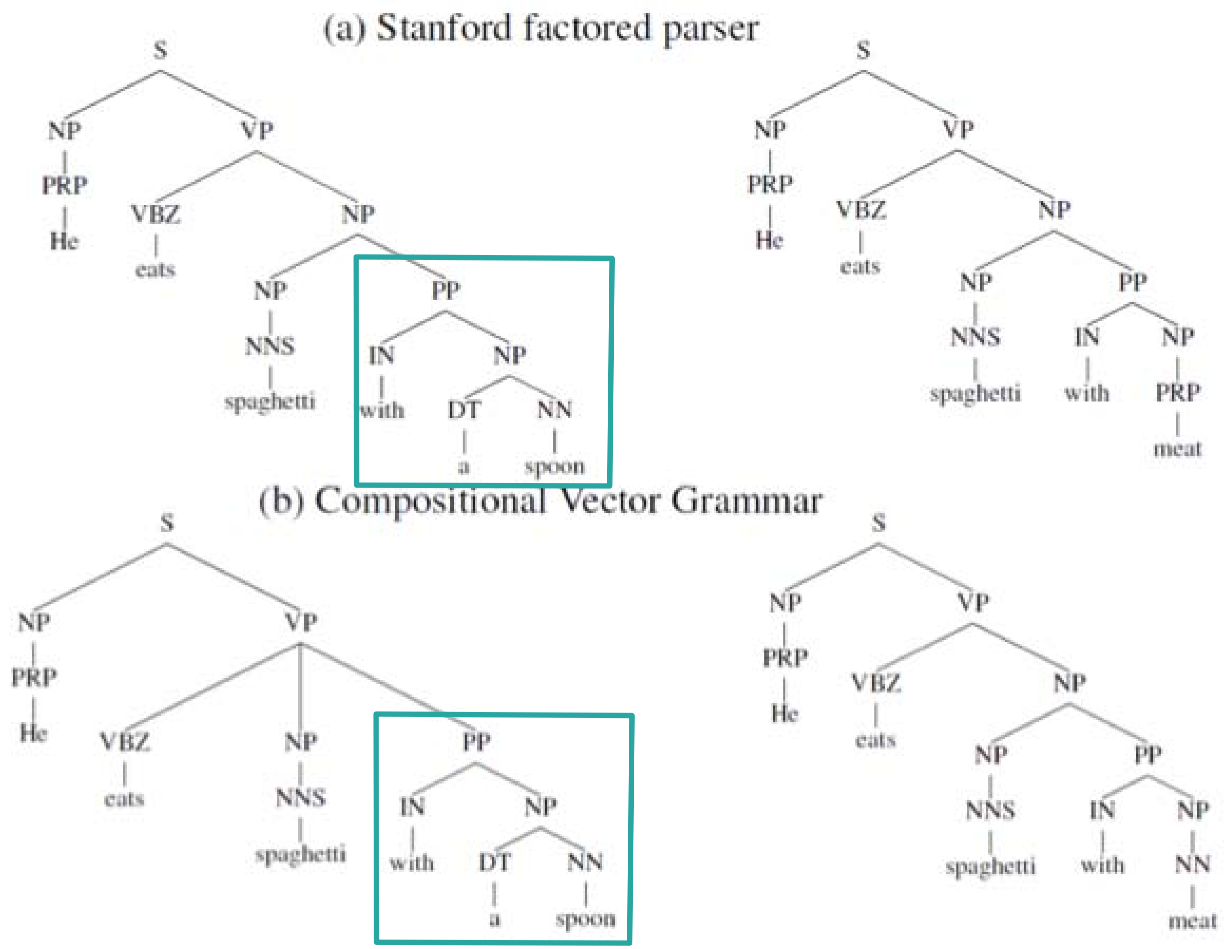
1. Sales rose more than 7% to $94.9 m. from $88.3 m.
2. Sales surged 40% to UNK b. yen from UNK b.



SU-RNN Analysis

- Can transfer semantic information from single related example
- Train sentences:
  - He eats spaghetti with a fork.
  - She eats spaghetti with pork.
- Test sentences
  - He eats spaghetti with a spoon.
  - He eats spaghetti with meat.

SU-RNN Analysis



Outline

- Motivation
- Recursive Neural Networks
  - Compositionality
  - Structure Prediction: Parsing
  - Backpropagation Through Structure
  - Minor twist leads to an improved Parser
- Matrix-Vector RNNs

Compositionality By Matrix-Vector RNNs

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

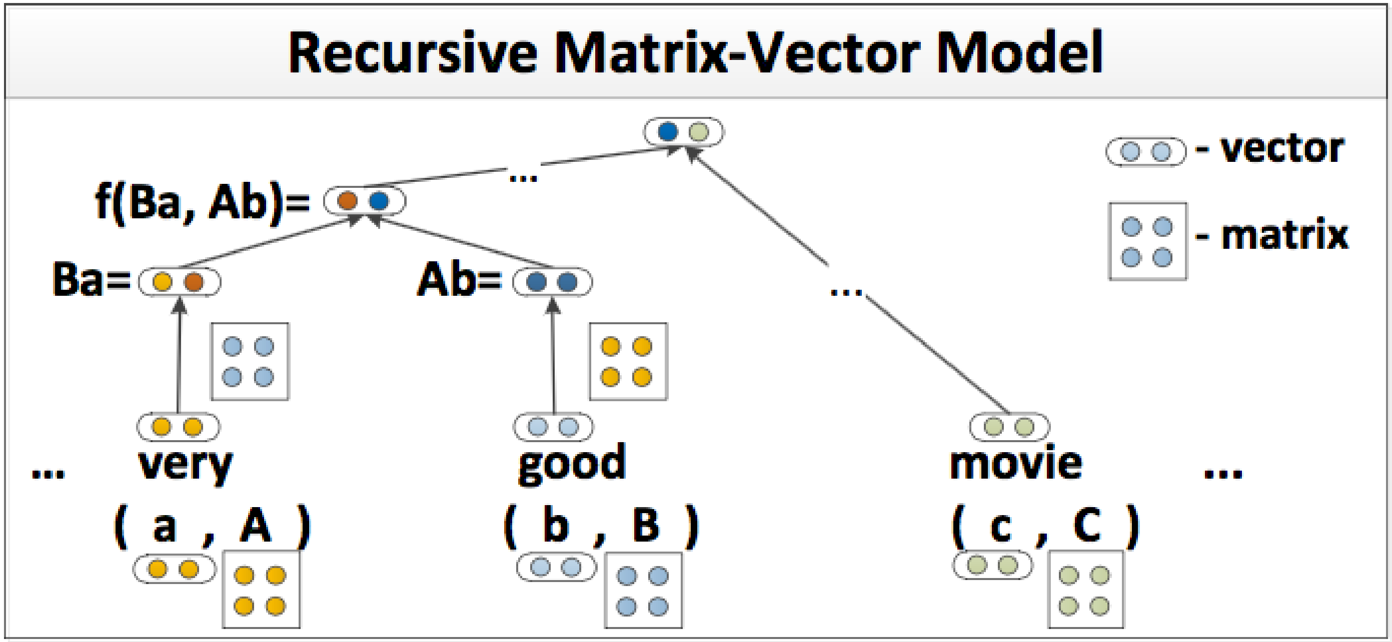
One way to make the composition function more powerful was by untying the weights *W*

But what if words act mostly as an operator, e.g. “very” in
very good

Proposal: A new composition function

Compositionality By Matrix-Vector RNNs

$$p = f_{A,B}(a,b) = f(Ba, Ab) = g\left(W \begin{bmatrix} Ba \\ Ab \end{bmatrix}\right) \qquad P = f_M(A,B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$$



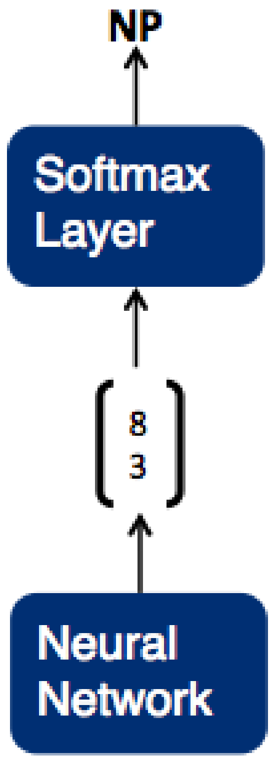
a: Captures the **meaning** of word “very”
A: Captures how it **modifies** the meaning of other words it combines with

Labeling tasks in RNN

- We can just use softmax classifier over the representation vector at root node:

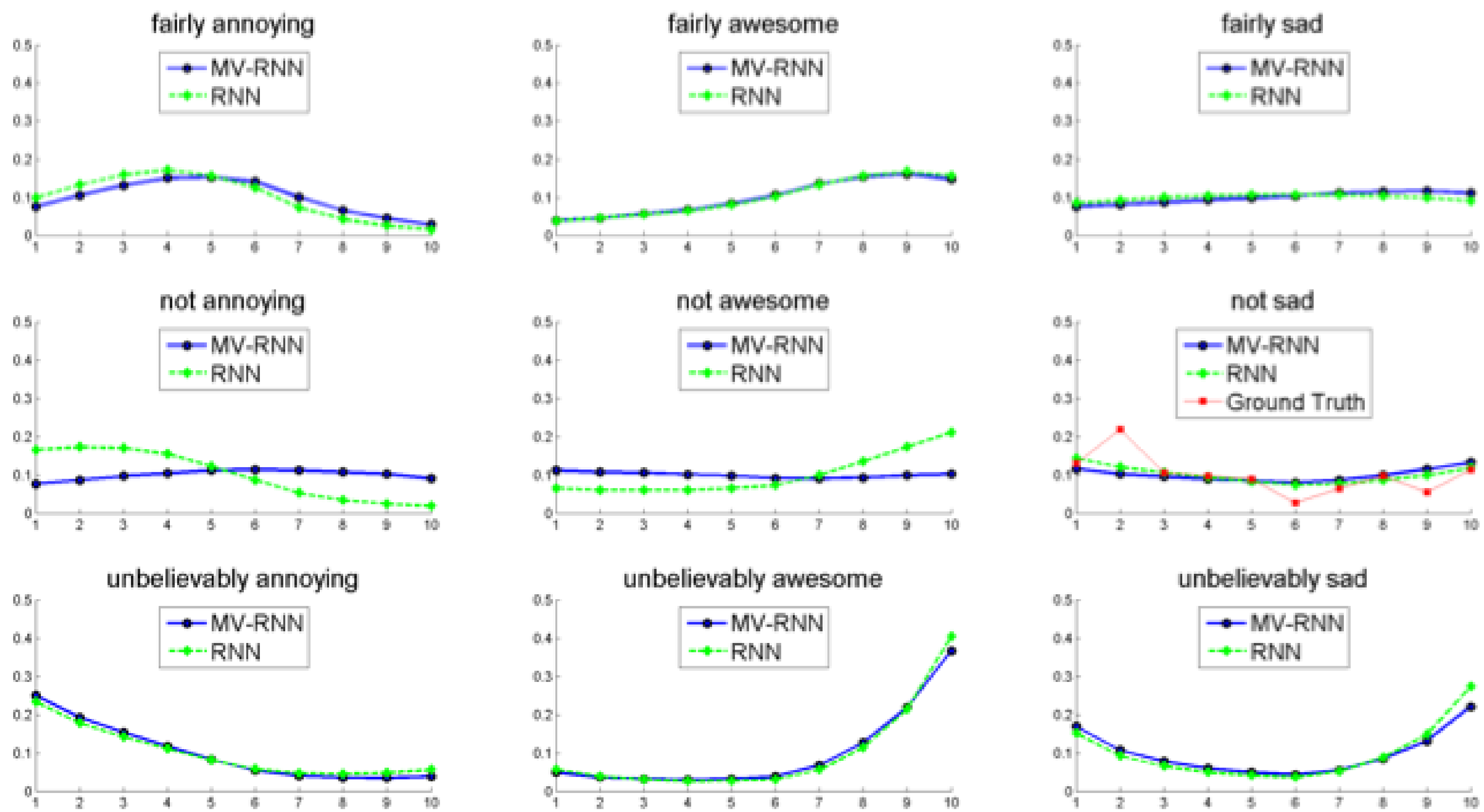
$$p(c|p) = softmax(Sp)$$

- Training similarly with standard cross-entropy error + scores



Predicting Sentiment Distributions of Adverb-Adjective Pairs (from IMDB)

Good example for non-linearity in language

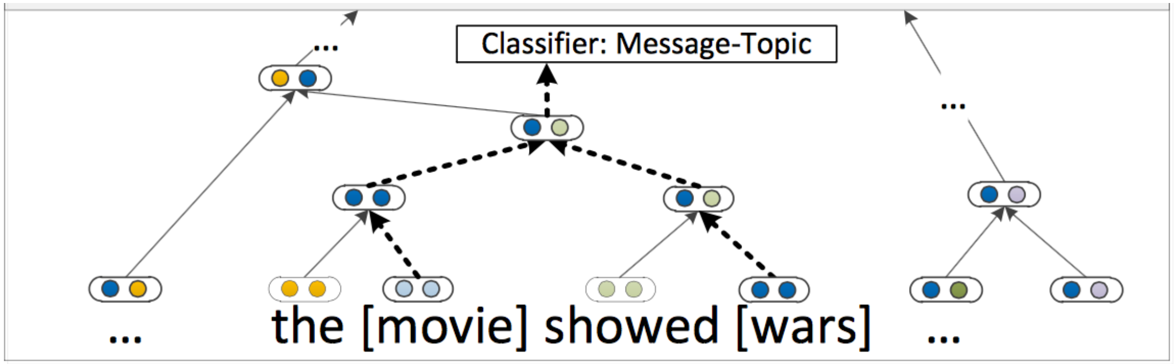


Hard movie review examples

S.	C.	Review sentence
1	√	The film is bright and flashy in all the right ways.
0	√	Not always too whimsical for its own good this strange hybrid of crime thriller, quirky character study, third-rate romance and female empowerment fantasy never really finds the tonal or thematic glue it needs.
0	√	Doesn't come close to justifying the hype that surrounded its debut at the Sundance film festival two years ago.
0	x	Director Hoffman, his writer and Kline's agent should serve detention.
1	x	A bodice-ripper for intellectuals.

Table 2: Hard movie review examples of positive (1) and negative (0) sentiment (S.) that of all methods only the MV-RNN predicted correctly (C: √) or could not classify as correct either (C: x).

Relationship Classification with MV-RNN



- SemEval-2010 Data
- 9 Relation Classes

Relationship	Sentence with labeled nouns for which to predict relationships
Cause-Effect(e2,e1)	Avian [influenza]$_{e1}$ is an infectious disease caused by type a strains of the influenza [virus]$_{e2}$.
Entity-Origin(e1,e2)	The [mother]$_{e1}$ left her native [land]$_{e2}$ about the same time and they were married in that city.
Message-Topic(e2,e1)	Roadside [attractions]$_{e1}$ are frequently advertised with [billboards]$_{e2}$ to attract tourists.
Product-Producer(e1,e2)	A child is told a [lie]$_{e1}$ for several years by their [parents]$_{e2}$ before he/she realizes that ...
Entity-Destination(e1,e2)	The accident has spread [oil]$_{e1}$ into the [ocean]$_{e2}$.
Member-Collection(e2,e1)	The siege started, with a [regiment]$_{e1}$ of lightly armored [swordsmen]$_{e2}$ ramming down the gate.
Instrument-Agency(e2,e1)	The core of the [analyzer]$_{e1}$ identifies the paths using the constraint propagation [method]$_{e2}$.
Component-Whole(e2,e1)	The size of a [tree]$_{e1}$ [crown]$_{e2}$ is strongly correlated with the growth of the tree.
Content-Container(e1,e2)	The hidden [camera]$_{e1}$, found by a security guard, was hidden in a business card-sized [leaflet box]$_{e2}$ placed at an unmanned ATM in Tokyo's Minato ward in early September.

References

1. Learning Continuous Phrase Representations and Syntactic Parsing with Recursive Neural Networks (Socher et al. NIPS *2010*)
2. Parsing Natural Scenes and Natural Language with Recursive Neural Networks (Socher et al. ICML 2011)
3. Semantic Compositionality thorugh Recursive Matrix-Vector Spaces (Socher et al. EMNLP 2012)
4. Parsing with Compositional Vector Grammars (Socher et al. ACL 2013)
5. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank (Socher et al. EMNLP 2013)

Backup Slides

Illustration Recursive NN: forwardProp

```
def forwardProp(self,node):
    # Recursion
    ...

    # This node's hidden activation
    node.h = np.dot(self.W,np.hstack([node.left.h, node.right.h])) + self.b
    # Relu
    node.h[node.h<0] = 0

    # Softmax
    node.probs = np.dot(self.Ws,node.h) + self.bs
    node.probs -= np.max(node.probs)
    node.probs = np.exp(node.probs)
    node.probs = node.probs/np.sum(node.probs)
```

Illustration Recursive NN: backwardProp

```
def backProp(self,node,error=None):
    # Softmax grad
    deltas = node.probs
    deltas[node.label] -= 1.0
    self.dWs += np.outer(deltas,node.h)
    self.dbs += deltas
    deltas = np.dot(self.Ws.T,deltas)

    # Add deltas from above
    if error is not None:
        deltas += error

    # f'(z) now:
    deltas *= (node.h != 0)

    # Update word vectors if leaf node:
    if node.isLeaf:
        self.dL[node.word] += deltas
        return

    # Recursively backprop
    if not node.isLeaf:
        self.dW += np.outer(deltas,np.hstack([node.left.h, node.right.h]))
        self.db += deltas
        # Error signal to children
        deltas = np.dot(self.W.T, deltas)
        self.backProp(node.left, deltas[:self.hiddenDim])
        self.backProp(node.right, deltas[self.hiddenDim:])
```

$$\boxed{\delta^{(l)} = \left((W^{(l)})^T \delta^{(l+1)}\right) \circ f'(z^{(l)}),}$$

$$\boxed{\frac{\partial}{\partial W^{(l)}} E_R = \delta^{(l+1)} (a^{(l)})^T + \lambda W^{(l)}}$$

Recursive Neural Tensor Networks (Advanced)

Motivation - Sentiment Detection

Most methods start with a bag of words
+ linguistic features/processing/lexica

But such methods (including tf-idf) can't distinguish:

Difficult!

+ white blood cells destroying an infection

– an infection destroying white blood cells

Motivation - Sentiment Detection

- For dataset of single sentence movie reviews (Pang and Lee, 2005) accuracy never reached above 80% for >7 years
- Harder cases require actual understanding of **negation and its scope** + other semantic effects

Motivation - Sentiment Detection

Examples from Movie Reviews Dataset:

Stealing Harvard doesn't care about cleverness, wit or any other kind of intelligent humor.

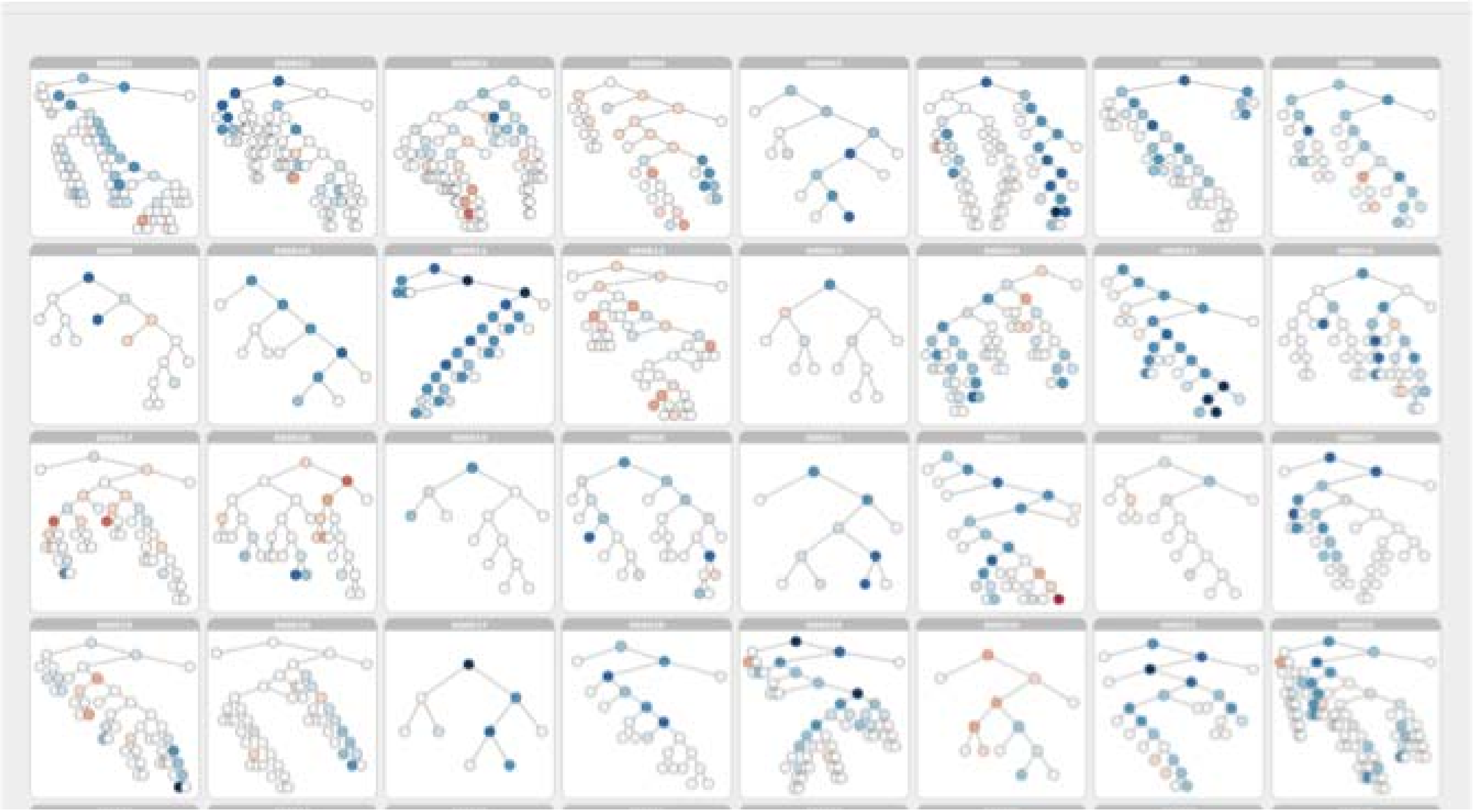
There are slow and repetitive parts but it has just enough spice to keep it interesting.

What is missing?

1. Compositional Training Data

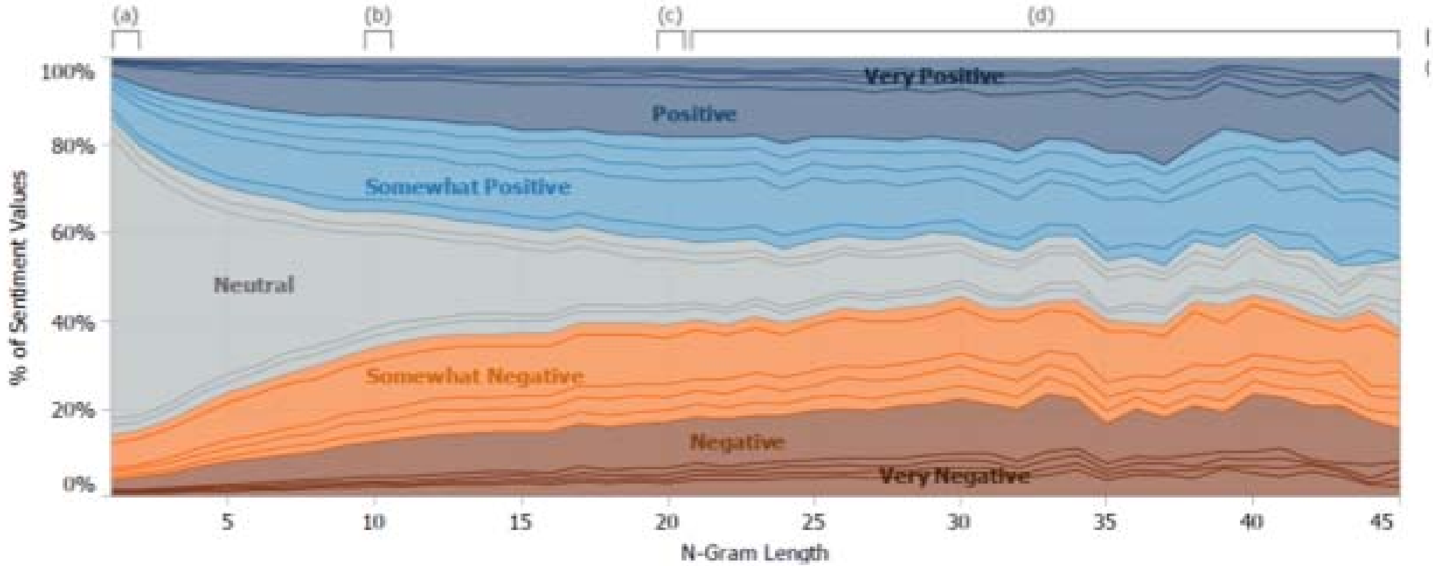
2. Better Compositional model

Solution 1: New Sentiment Treebank



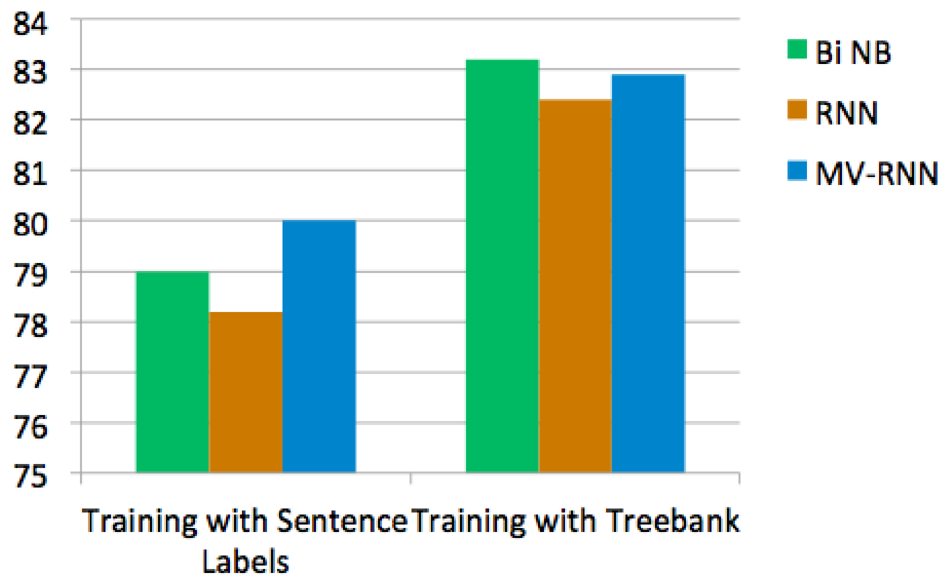
Solution 1: New Sentiment Treebank

- Parse trees of 11,855 sentences
- 215,154 phrases with labels
- Allows training and evaluating with compositional information



Using Better Dataset Helped All Models

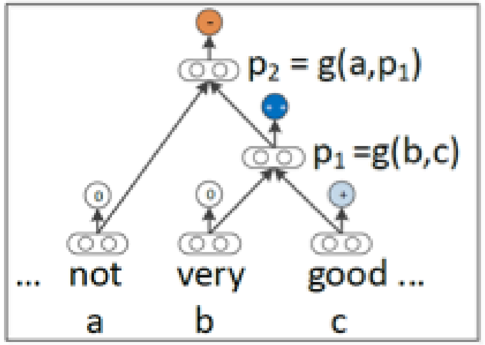
- Binary(pos/neg) full sentence classification

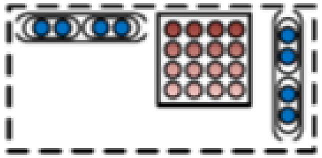


- Yet, hard negation cases are still mostly incorrect
- Better/stronger compositional model

Solution 2: New Compositional Model

- Recursive Neural Tensor Network
- More expressive than previous RNNs
- Idea: Allow more interactions of vectors

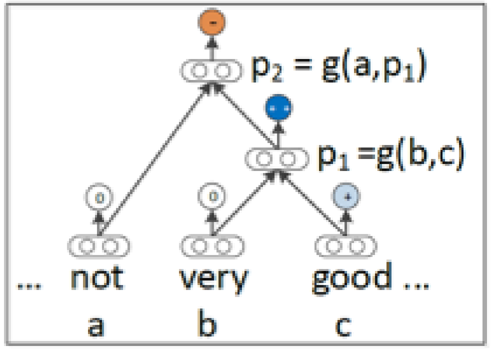


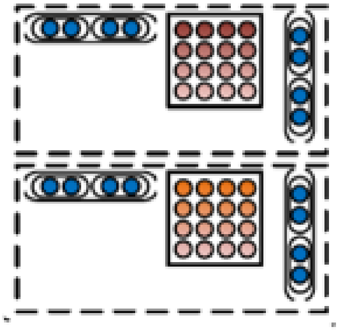


$$\begin{bmatrix} b \\ c \end{bmatrix}^T V \begin{bmatrix} b \\ c \end{bmatrix}$$

Solution 2: New Compositional Model

- Recursive Neural Tensor Network

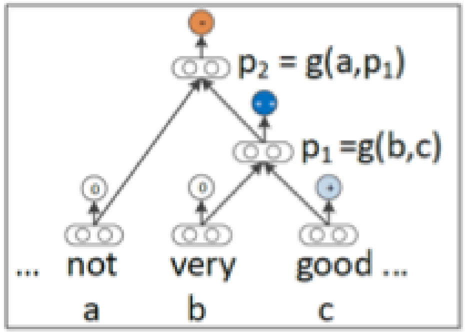


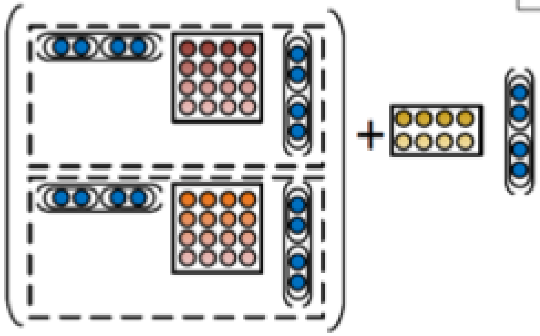


$$\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:2]} \begin{bmatrix} b \\ c \end{bmatrix}$$

Solution 2: New Compositional Model

- Recursive Neural Tensor Network

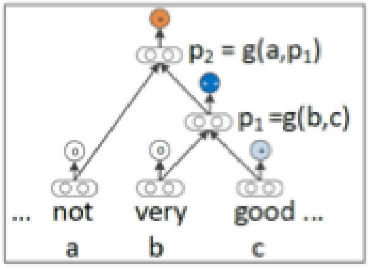


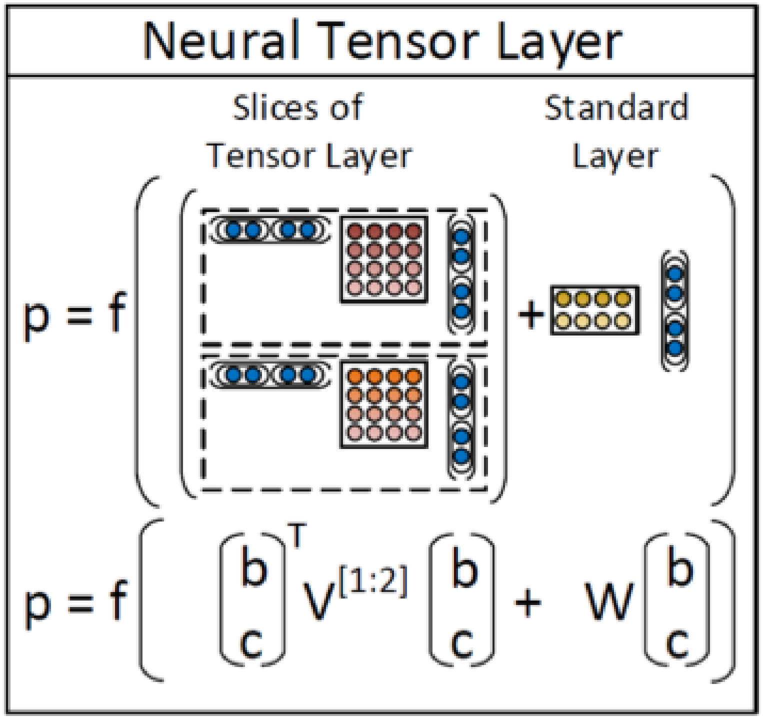


$$\begin{pmatrix} b \\ c \end{pmatrix}^T V^{[1:2]} \begin{pmatrix} b \\ c \end{pmatrix} + W \begin{pmatrix} b \\ c \end{pmatrix}$$

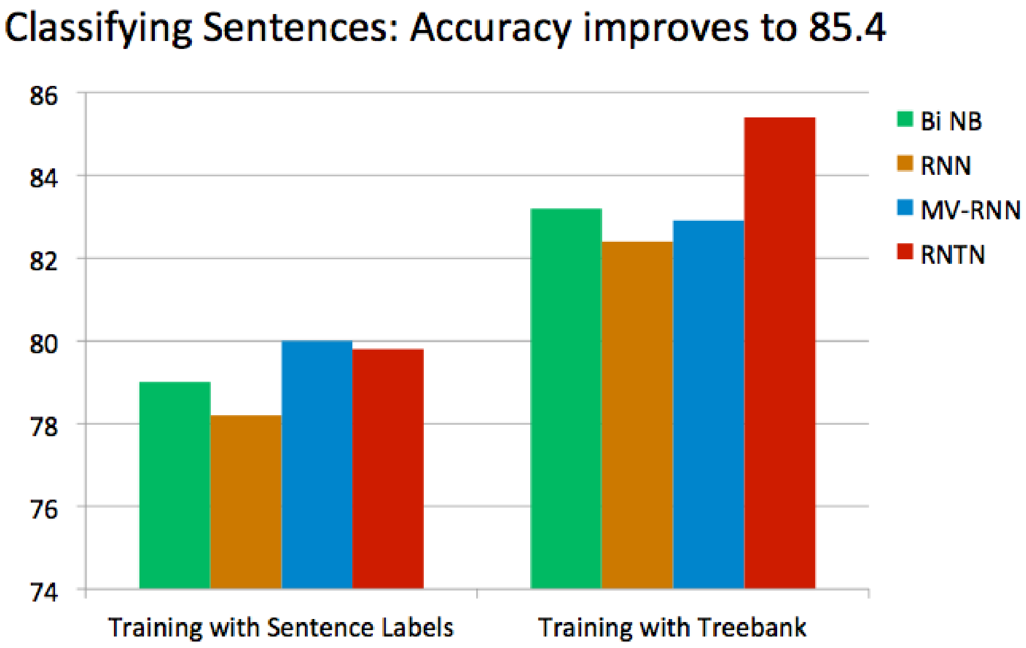
Recursive Neural Tensor Network

Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank
Socher et al. 2013





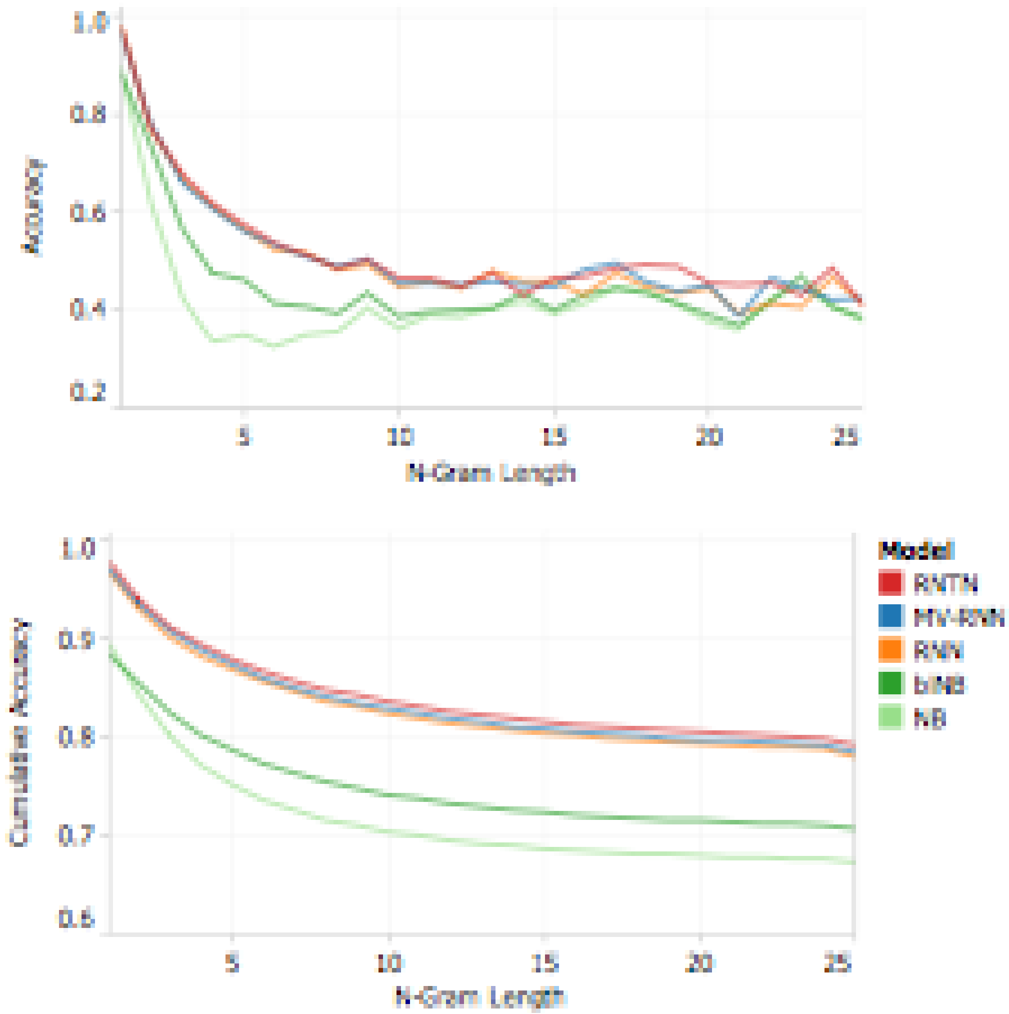
Positive/Negative Results on Treebank



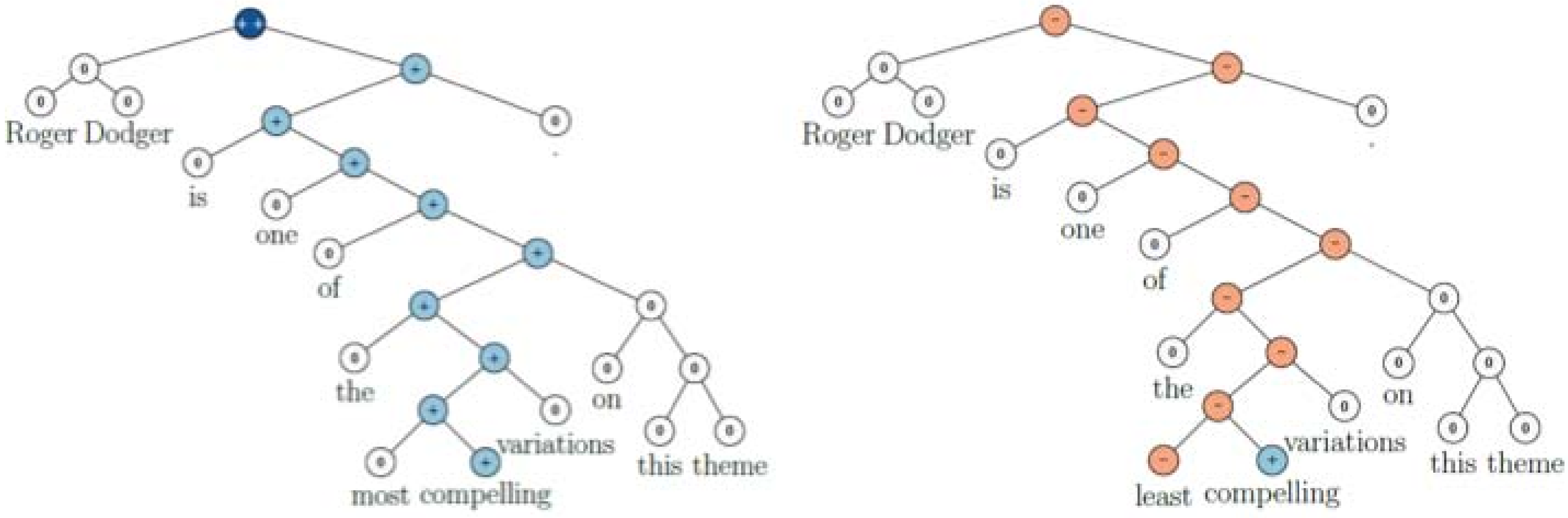
Model	Fine-grained		Positive/Negative	
	All	Root	All	Root
NB	67.2	41.0	82.6	81.8
SVM	64.3	40.7	84.6	79.4
BiNB	71.0	41.9	82.7	83.1
VecAvg	73.3	32.7	85.1	80.1
RNN	79.0	43.2	86.1	82.4
MV-RNN	78.7	44.4	86.8	82.9
RNTN	**80.7**	**45.7**	**87.6**	**85.4**

Root: Sentence Level

Fina Grained Sentiment Classification Results



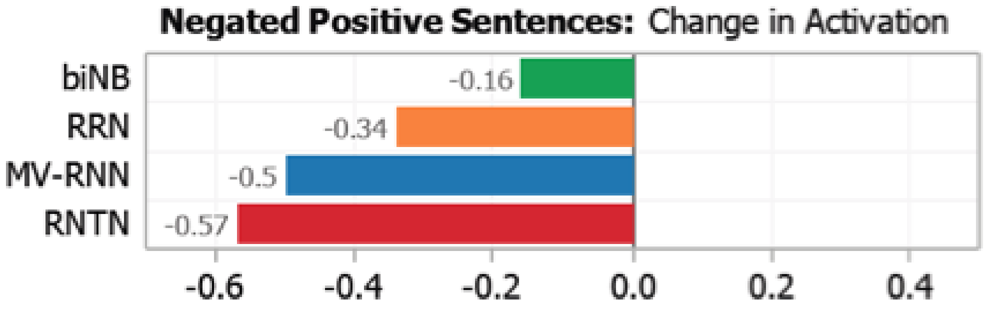
Negation Results



Negation Results

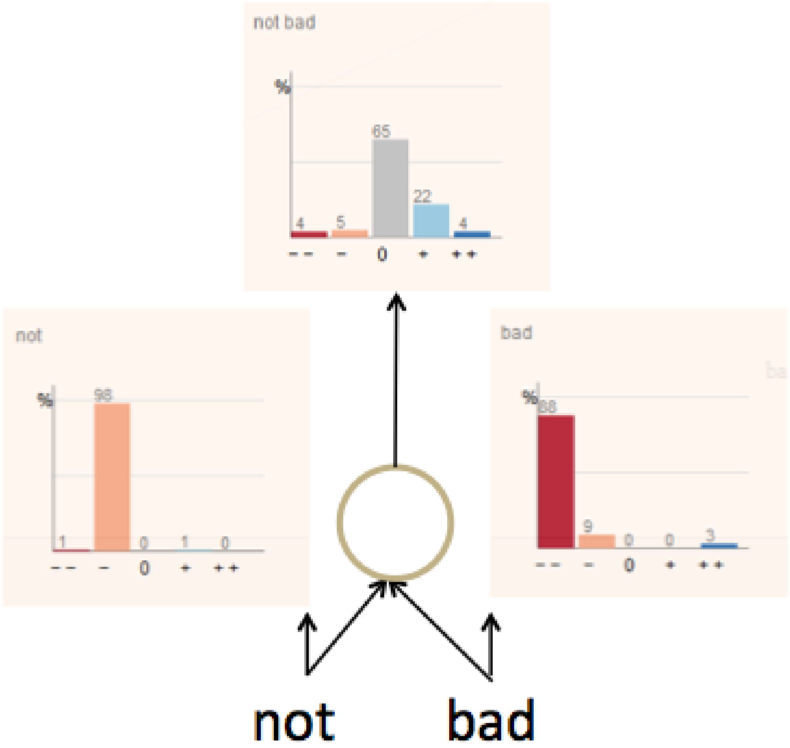
- Most methods capture that negation often makes things more negative (See Potts, 2010)
- Analysis on negation dataset
- Accuracy:

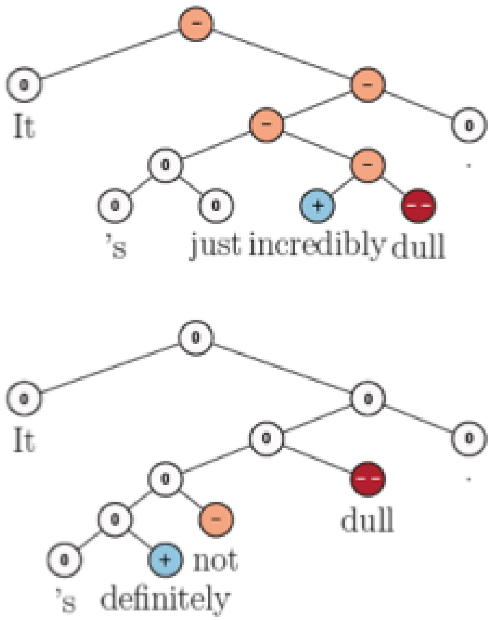
	Negated Positive
biNB	19.0
RNN	33.3
MV-RNN	52.4
RNTN	**71.4**



Results on Negating Negatives

- But how about negating negatives?
- No flips, but positive activation should increase!





Results on Negating Negatives

- Evaluation: Positive activation should increase

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**	**81.8**

Negated Negative Sentences: Change in Activation