CS 291A: Deep Learning for NLP

Neural Networks: Recursive Neural Networks

William Wang
UCSB Computer Science
william@cs.ucsb.edu

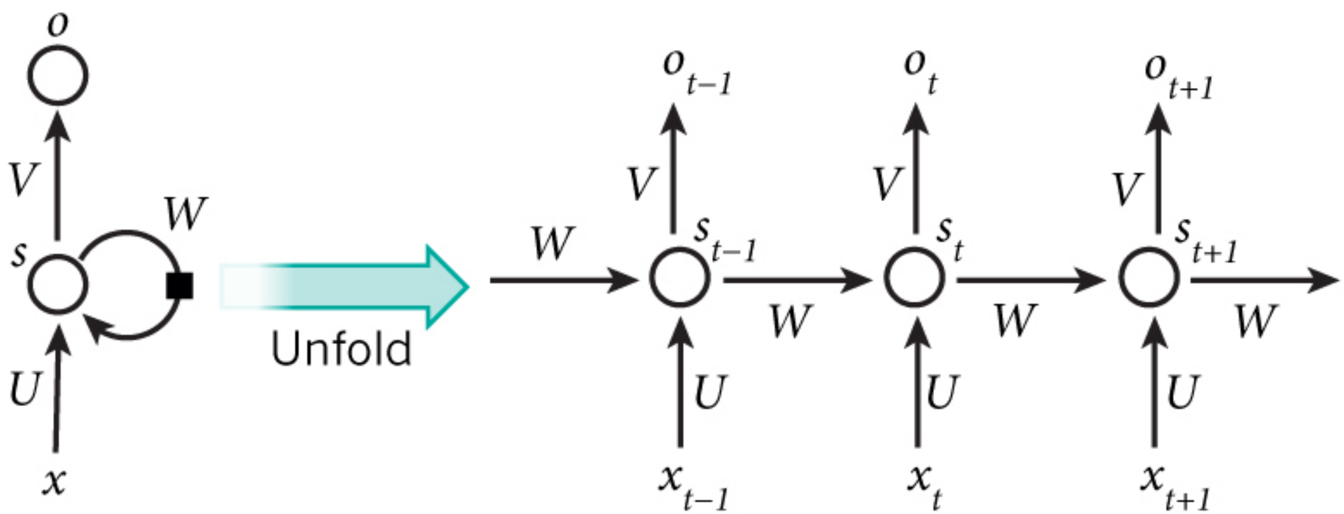
Slides adapted from Y. V. Chen and R. Socher.

Announcements

Project proposal

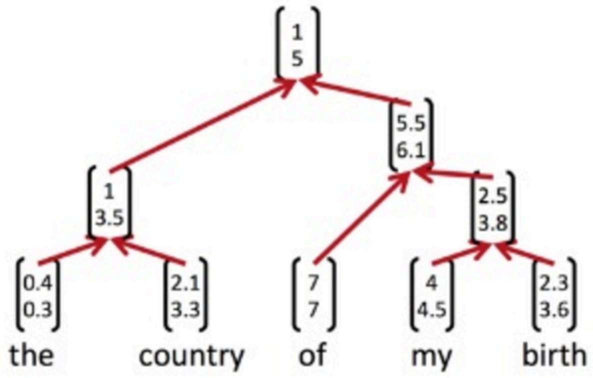
1. If you have not submitted your project proposal, please let me know.
2. Most proposals are very creative, and please come to my office hour next week if you need feedback.
3. Need to be more concrete on your technical approach.

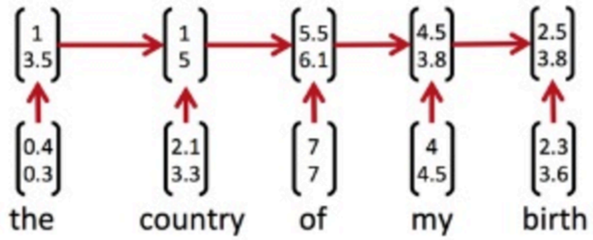
Recursive vs. Recurrent



Is this recursive or recurrent?

Recursive vs. Recurrent





Sequence Modeling

Idea: aggregate the meaning from all words into a vector

→ ***Compositionality***

Method:

- Basic combination: average, sum
- Neural combination:
  - ✓Recursive neural network (RvNN)
  - ✓Recurrent neural network (RNN)
  - ✓Convolutional neural network (CNN)

$$\begin{array}{ll}
 & \overbrace{\hphantom{\begin{bmatrix} 0.2 & 0.6 & 0.3 & \cdots & 0.4 \end{bmatrix}}}^{N\text{-dim}} \\
\text{this} & \begin{bmatrix} 0.2 & 0.6 & 0.3 & \cdots & 0.4 \end{bmatrix} \\
\text{idea} & \begin{bmatrix} 0.9 & 0.8 & 0.1 & \cdots & 0.1 \end{bmatrix} \\
\text{is} & \begin{bmatrix} 0.1 & 0.3 & 0.1 & \cdots & 0.7 \end{bmatrix} \\
\text{amazing} & \begin{bmatrix} 0.5 & 0.0 & 0.6 & \cdots & 0.4 \end{bmatrix}
\end{array}$$

How to compute $\vec{x} = \begin{bmatrix} x_1 & x_2 & x_3 & \cdots & x_N \end{bmatrix}$

Recursive Neural Network

From Words to Phrases

Recursive Neural Network

Idea: leverage the linguistic knowledge (syntax) for combining multiple words into phrases

Assumption: language is described recursively

Related Work for RvNN

Pollack (1990): Recursive auto-associative memories

Previous Recursive Neural Networks work by Goller & Küchler (1996), Costa et al. (2003) assumed fixed tree structure and used one-hot vectors.

Hinton (1990) and Bottou (2011): Related ideas about recursive models and recursive operators as smooth versions of logic operations

Outline

Property

- Compositionality
- Recursion Assumption

Network Architecture and Definition

- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications

- Parsing
- Paraphrase Detection
- Sentiment Analysis

Outline

Property
- **Compositionality**
- Recursion Assumption

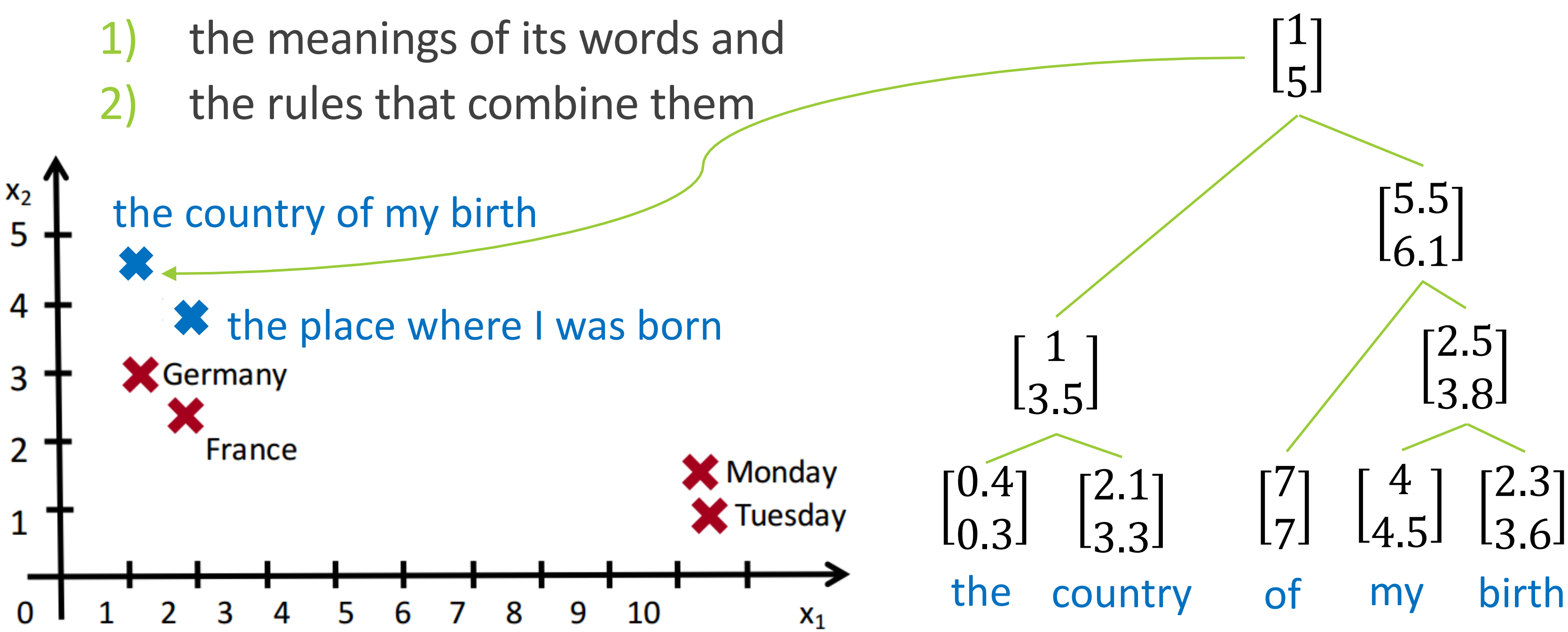
Network Architecture and Definition
- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications
- Parsing
- Paraphrase Detection
- Sentiment Analysis

Phrase Mapping

Principle of "Compositionality"

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



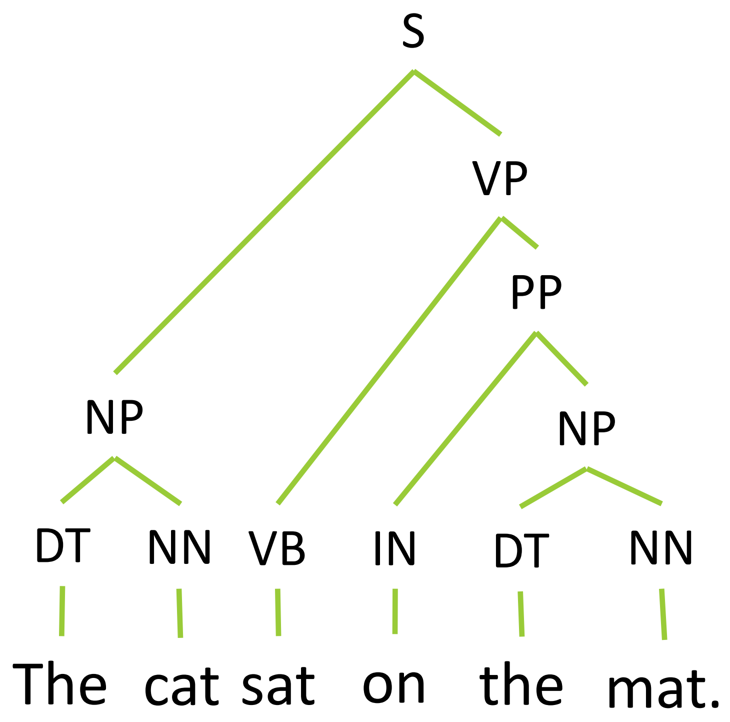
Idea: jointly learn parse trees and compositional vector representations

Sentence Syntactic Parsing

Parsing is a process of analyzing a string of symbols

Parsing tree conveys

1) Part-of-speech for each word
2) Phrases
3) Relationships



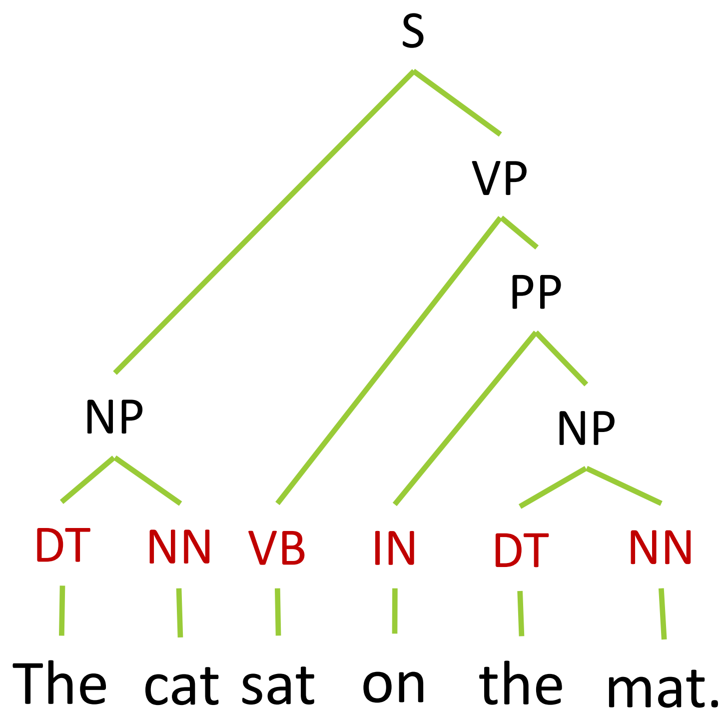
(NN = noun, VB = verb, DT = determiner, IN = Preposition)

Sentence Syntactic Parsing

Parsing is a process of analyzing a string of symbols

Parsing tree conveys

1) **Part-of-speech for each word**
2) Phrases
3) Relationships



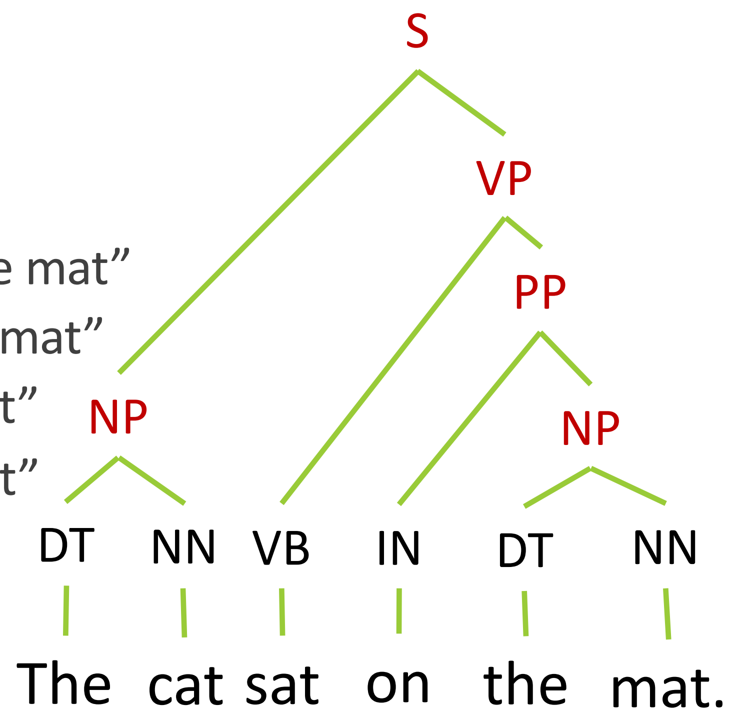
(NN = noun, VB = verb, DT = determiner, IN = Preposition)

Sentence Syntactic Parsing

Parsing is a process of analyzing a string of symbols

Parsing tree conveys

1) Part-of-speech for each word
2) **Phrases**
   - Noun phrase (NP): “the cat”, “the mat”
   - Preposition phrase (PP): “on the mat”
   - Verb phrase (VP): “sat on the mat”
   - Sentence: “the cat sat on the mat”
3) Relationships



(NN = noun, VB = verb, DT = determiner, IN = Preposition)

Sentence Syntactic Parsing

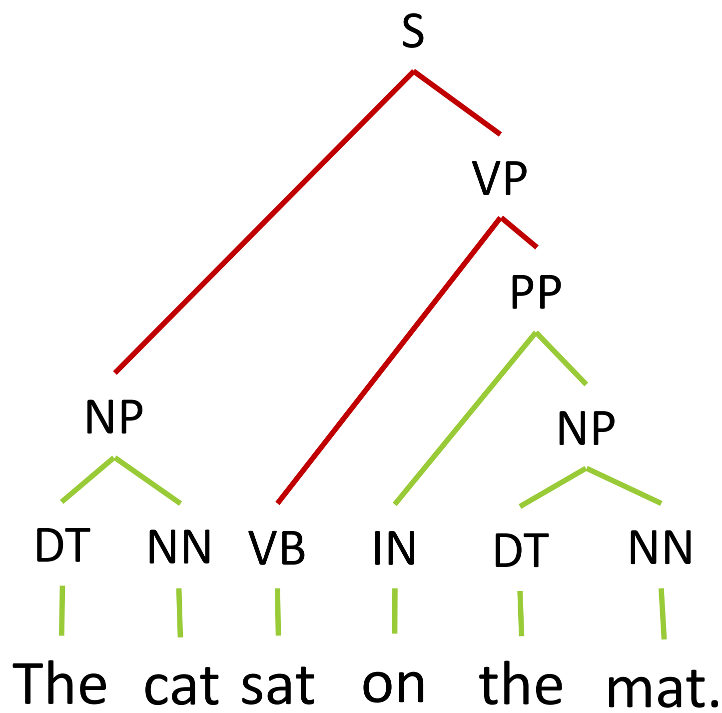
Parsing is a process of analyzing a string of symbols

Parsing tree conveys

1) Part-of-speech for each word
2) Phrases
3) **Relationships**

subject verb modifier_of_place

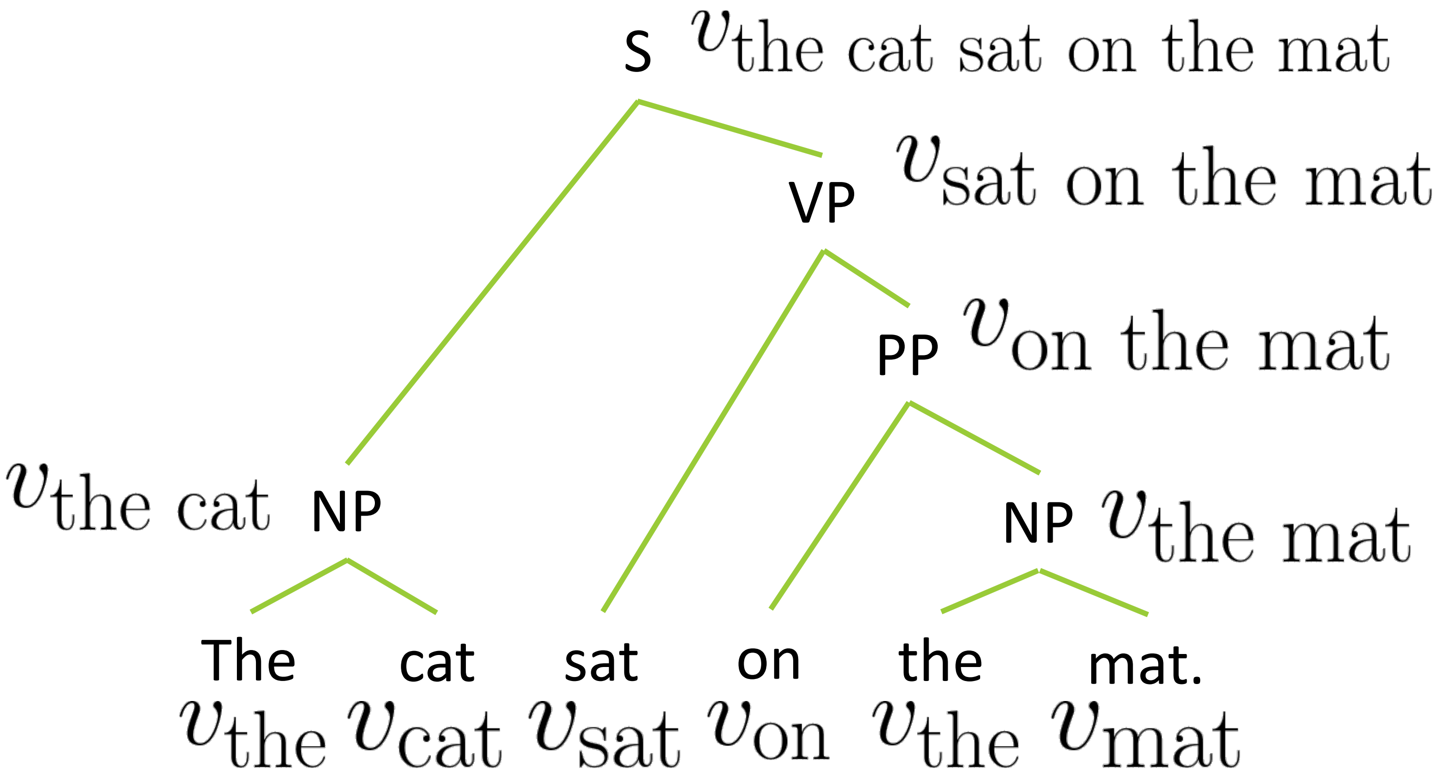
- "the cat" is the subject of "sat"
- "on the mat" is the place modifier of "sat"



(NN = noun, VB = verb, DT = determiner, IN = Preposition)

Learning Structure & Representation

Vector representations incorporate the meaning of words and their compositional structures



Outline

Property

- Syntactic Compositionality
- **Recursion Assumption**

Network Architecture and Definition

- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications

- Parsing
- Paraphrase Detection
- Sentiment Analysis

Recursion Assumption

Are languages recursive? debatable

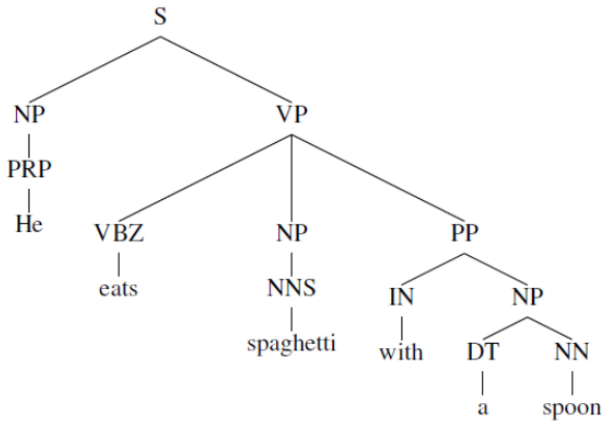
Recursion helps describe natural language

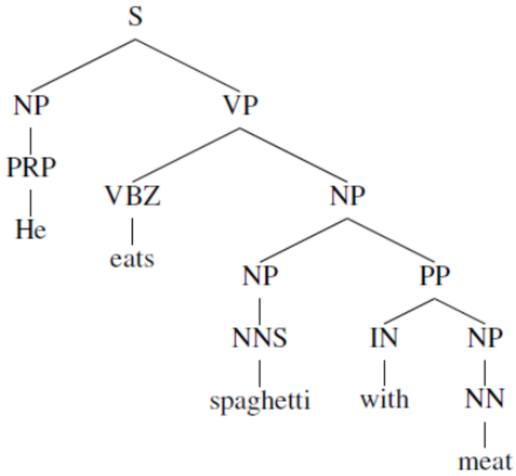
- Ex. “the church which has nice windows”, a noun phrase containing a relative clause that contains a noun phrases
- NP → NP PP

Recursion Assumption

Characteristics of recursion

1. Helpful in disambiguation





2. Helpful for some tasks 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
3. Works better for some tasks to use grammatical tree structure

Language recursion is still up to debate

Outline

Property
- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition
- **Standard Recursive Neural Network**
  - **Weight-Tied**
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications
- Parsing
- Paraphrase Detection
- Sentiment Analysis

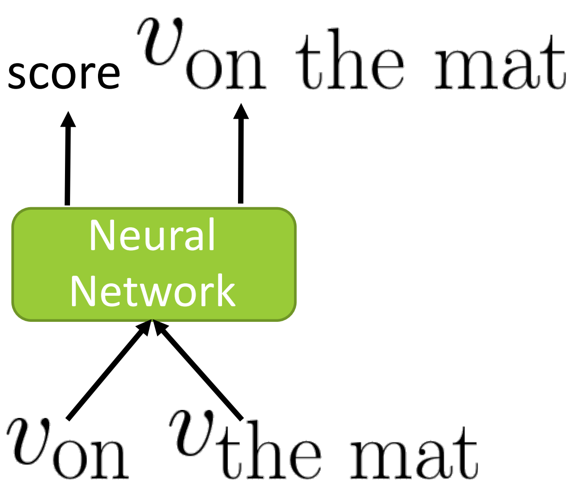
Recursive Neural Network Architecture

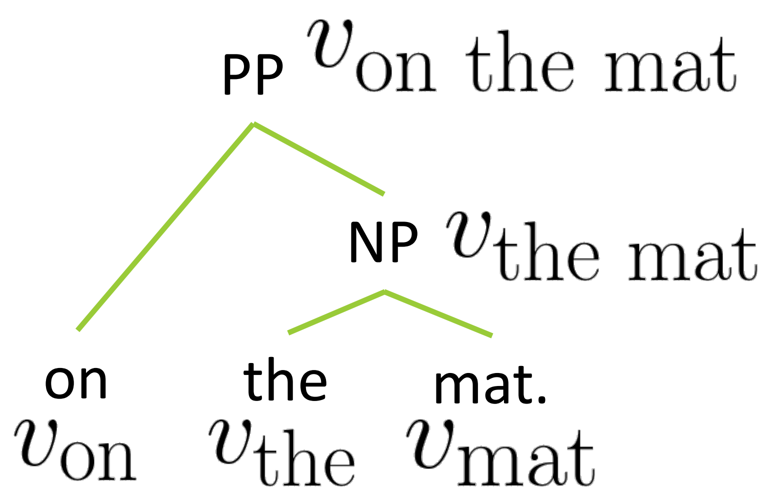
A network is to predict the vectors along with the structure

- Input: two candidate children's vector representations
- Output:

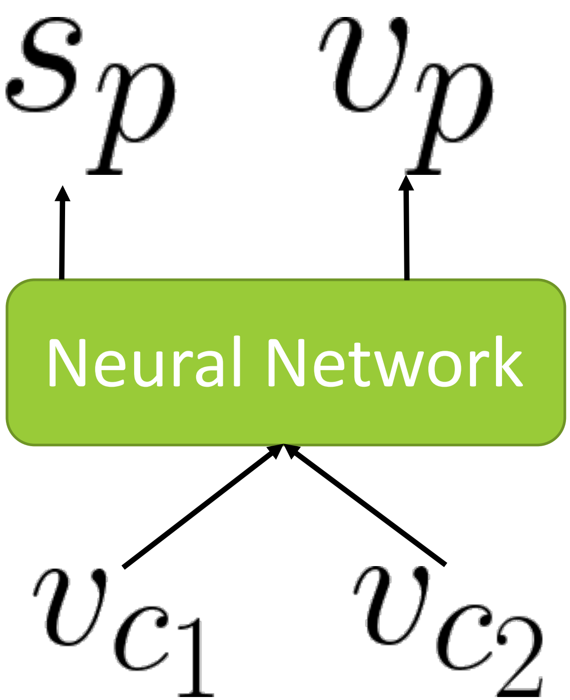
  1) vector representations for the merged node

  2) score of how plausible the new node would be





Recursive Neural Network Definition

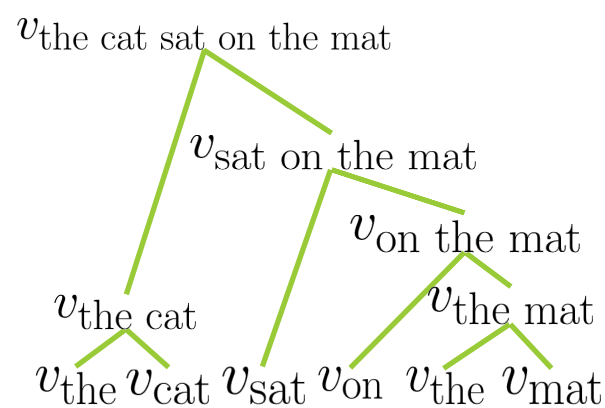


1) vector representations for the merged node

$$v_p = \sigma\left(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b\right)$$

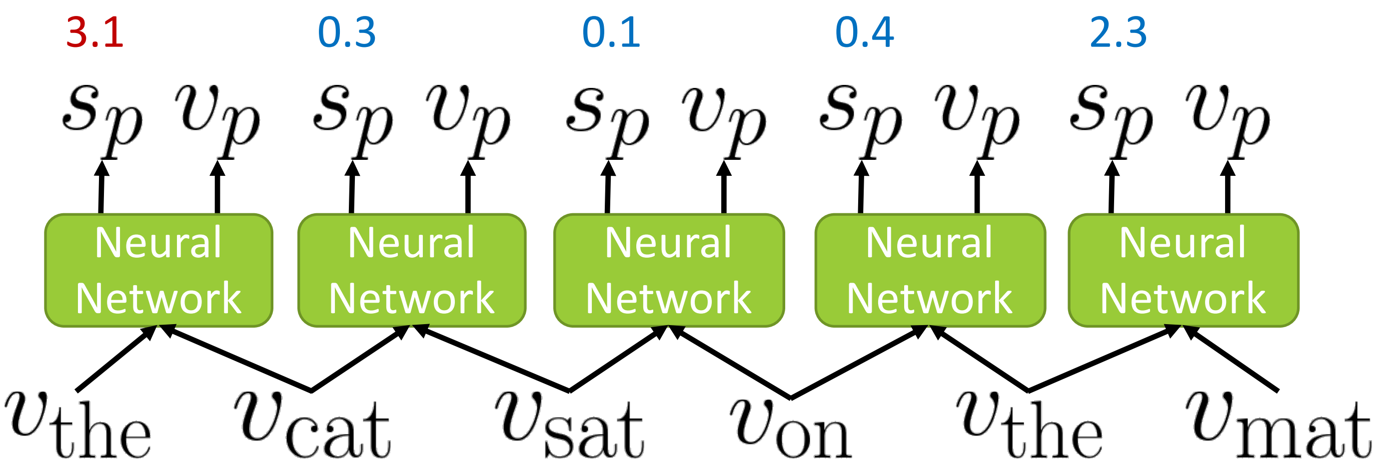
2) score of how plausible the new node would be

$$s_p = U^T v_p$$

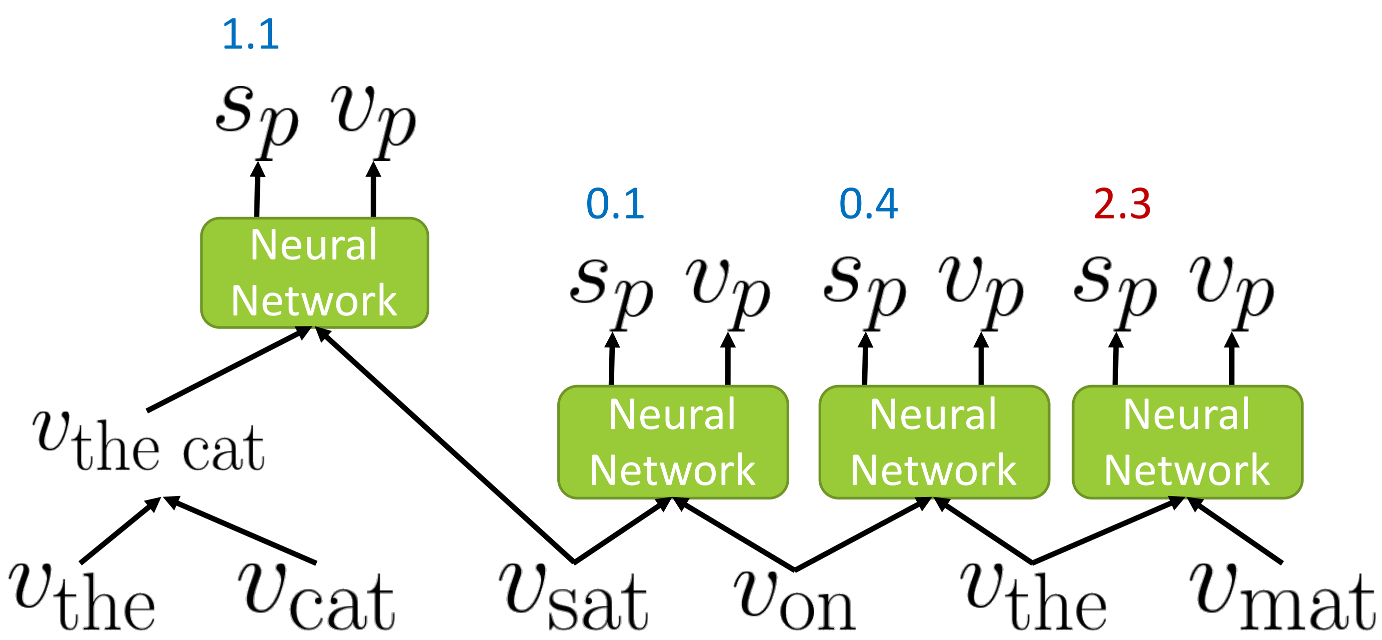


same W parameters at all nodes of the tree
→ weight-tied

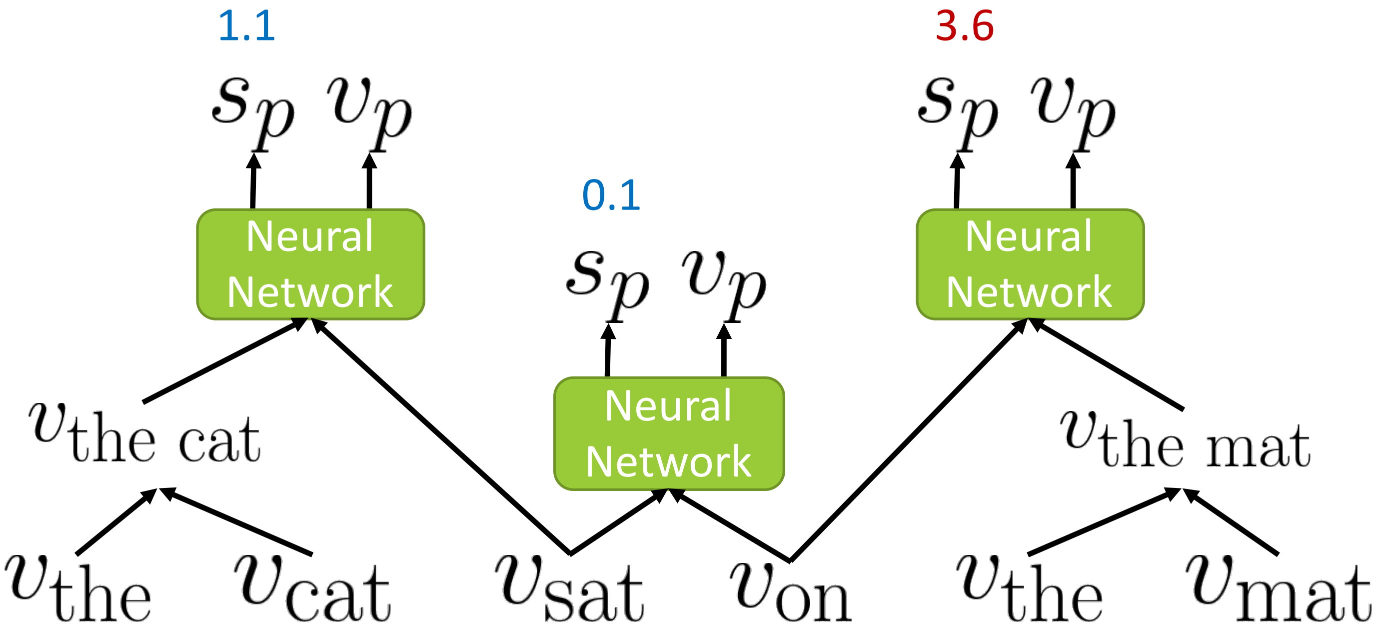
Sentence Parsing via RvNN



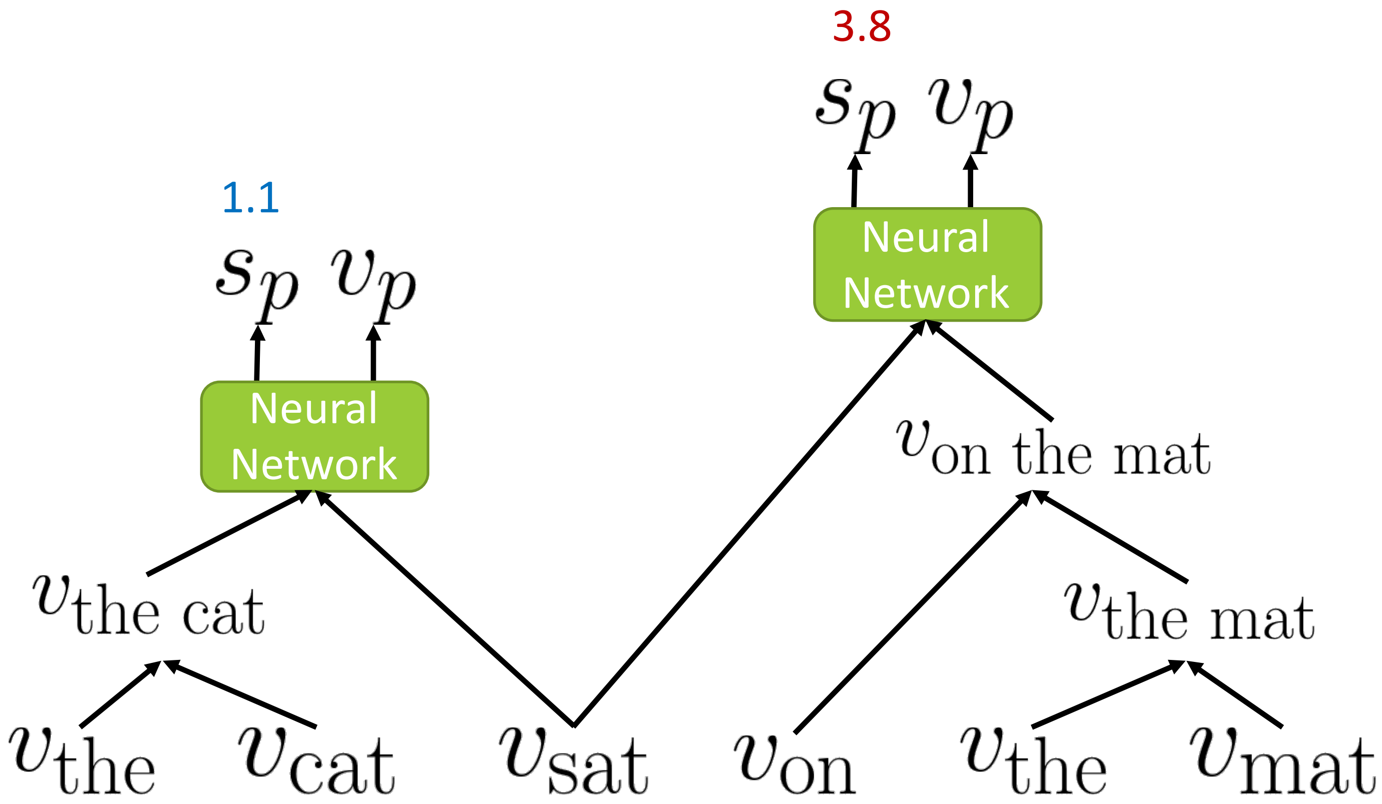
Sentence Parsing via RvNN



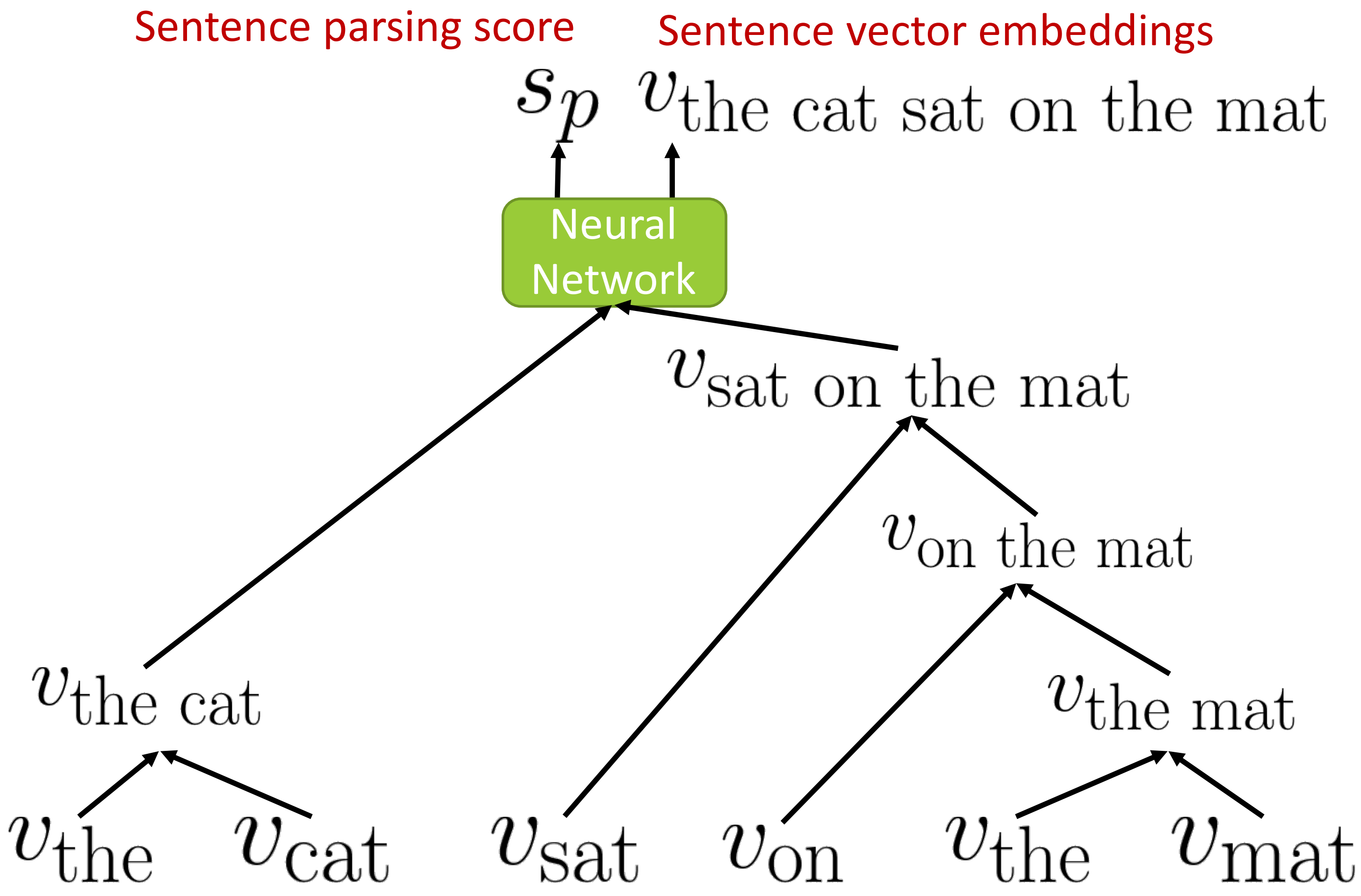
Sentence Parsing via RvNN



Sentence Parsing via RvNN



Sentence Parsing via RvNN



Backpropagation through Structure

Principally the same as general backpropagation (Goller& Küchler, 1996)

$$\frac{\partial C(\theta)}{\partial w_{ij}^l} = \boxed{\frac{\partial C(\theta)}{\partial z_i^l}} \boxed{\frac{\partial z_i^l}{\partial w_{ij}^l}}$$

$$\frac{\partial C(\theta)}{\partial z_i^l} = \boxed{\delta_i^l}$$

$$\frac{\partial z_i^l}{\partial w_{ij}^l} = \boxed{\begin{cases} a_j^{l-1} & l > 1 \\ x_j & l = 1 \end{cases}}$$

Backward Pass

$$\delta^L = \sigma'(z^L) \odot \nabla C(y)$$

$$\delta^{L-1} = \sigma'(z^{L-1}) \odot (W^L)^T \delta^L$$

$$\vdots$$

$$\delta^l = \sigma'(z^l) \odot (W^{l+1})^T \delta^{l+1}$$

$$\vdots$$

Forward Pass

$$z^1 = W^1 x + b^1 \qquad a^1 = \sigma(z^1)$$

$$\vdots$$

$$z^l = W^l a^{l-1} + b^l \quad a^l = \sigma(z^l)$$

$$\vdots$$

Three differences

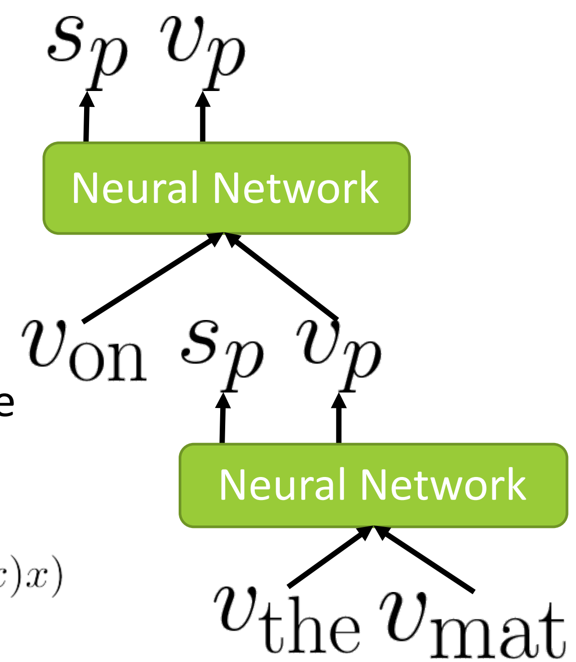
① Sum derivatives of W from all nodes
② Split derivatives at each node
③ Add error messages from parent + node itself

1) Sum derivatives of W from all nodes

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

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

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

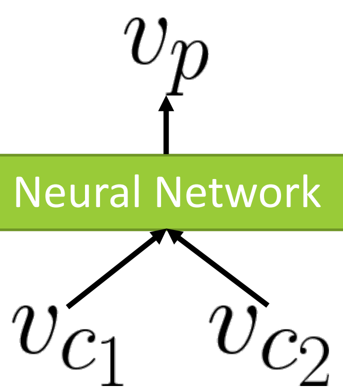


$$v_p = \sigma\left(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b\right)$$

2) Split derivatives at each node

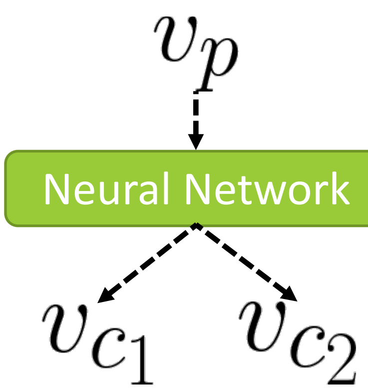
During forward propagation, the parent node is computed based on two children

$$v_p = \sigma(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b)$$



During backward propagation, the errors should be computed wrt each of them

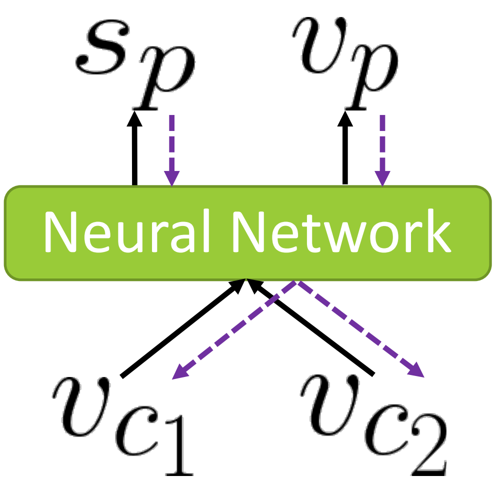
$$\delta_{p \to c_1 c_2} = [\delta_{p \to c_1} \delta_{p \to c_2}]$$



3) Add error messages

For each node, the error message is compose of

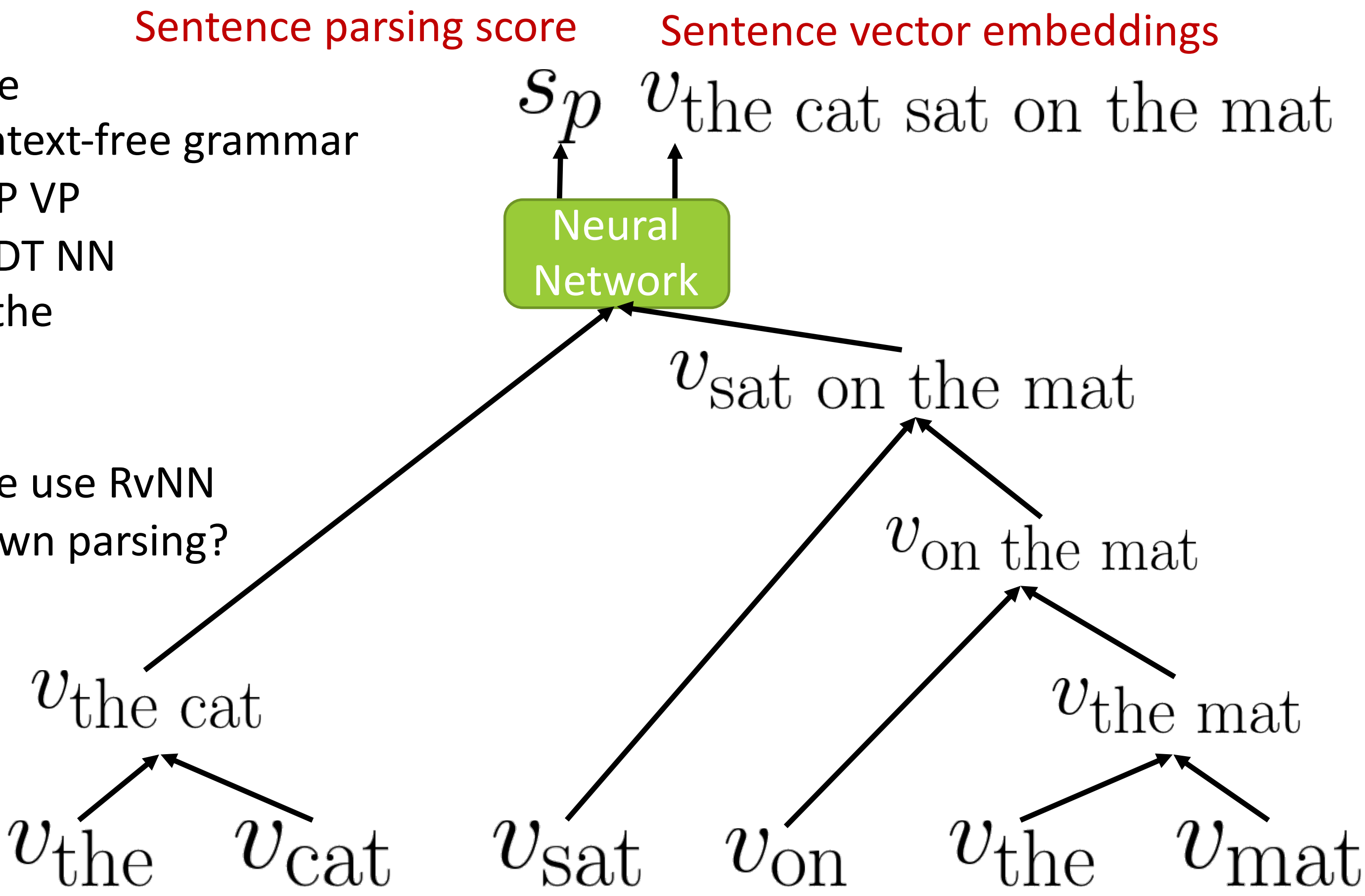
- Error propagated from parent
- Error from the current node



Group Discussion:
Top-Down Sentence Parsing via RvNN

Assume we
have a context-free grammar
E.g. S -> NP VP
NP -> DT NN
DT -> the
....

How do we use RvNN
for top-down parsing?



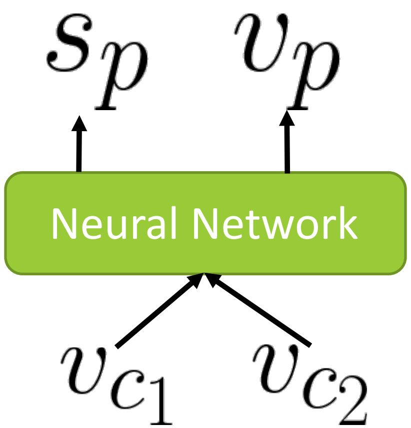
Outline

Property
- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition
- **Standard Recursive Neural Network**
  - Weight-Tied
  - **Weight-Untied**
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

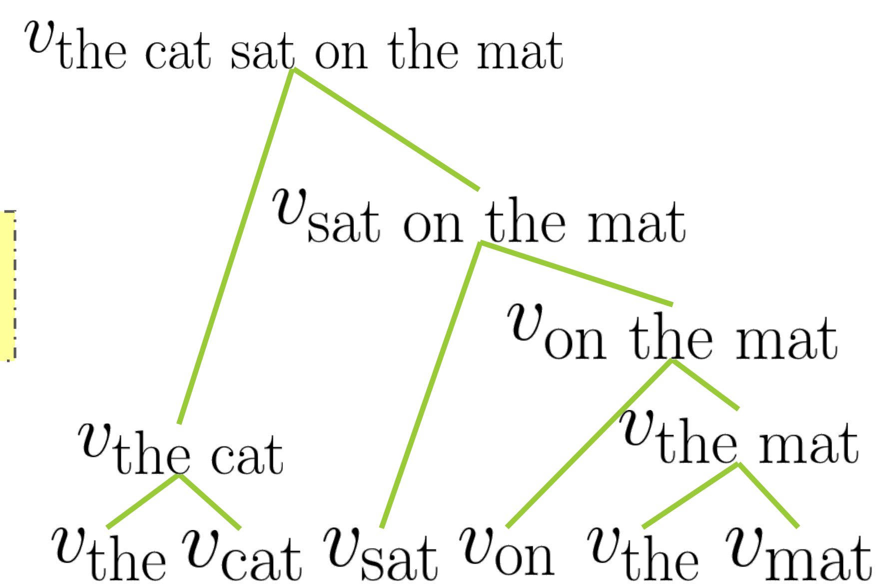
Applications
- Parsing
- Paraphrase Detection
- Sentiment Analysis

Composition Matrix W



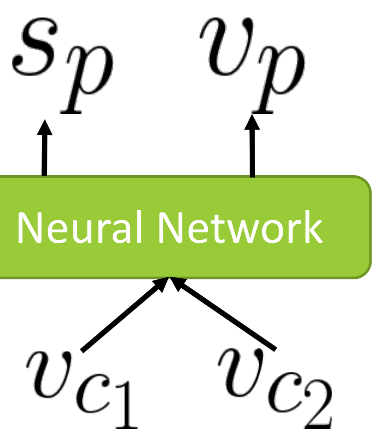
$$v_p = \sigma\left(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b\right)$$

$$s_p = U^T v_p$$



Issue: using the same network W for different compositions

Syntactically Untied RvNN

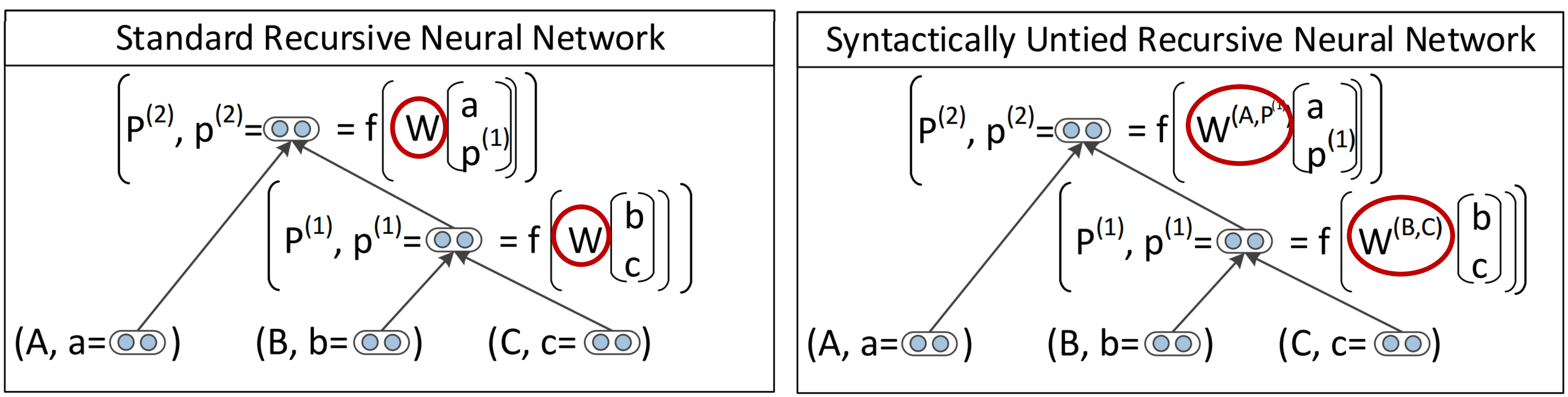


Idea: the composition function is conditioned on the syntactic categories

Benefit

- Composition function are syntax-dependent
- Allows different composition functions for word pairs, e.g. Adv + AdjP, VP + NP

Issue: speed due to many candidates



Compositional Vector Grammar

Compute score only for a subset of trees coming from a simpler, faster model (Socher et al, 2013)

- Prunes very unlikely candidates for speed
- Provides coarse syntactic categories of the children for each beam candidate

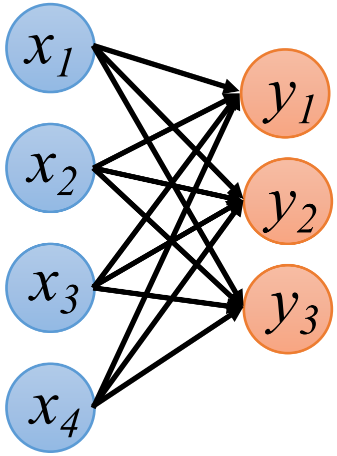
Probability context-free grammar (PCFG) helps decrease the search space

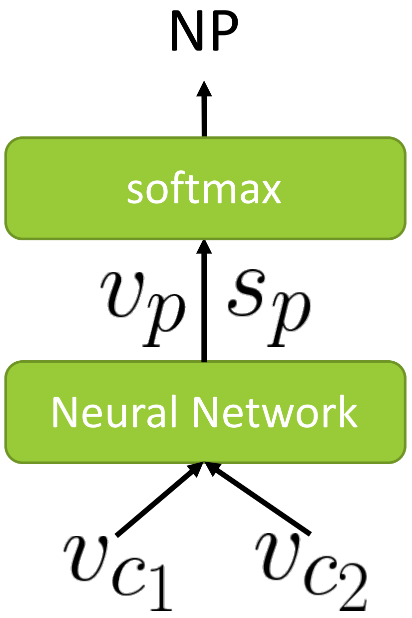
Labels for RvNN

The score can be passed through a softmax function to compute the probability of each category

$$p(\text{class} \mid v_p) = \text{softmax}(s_p)$$

$$\text{softmax}(f)_i = \frac{\exp(f_i)}{\sum_j \exp(f_j)}$$





Softmax loss → cross-entropy error for optimization

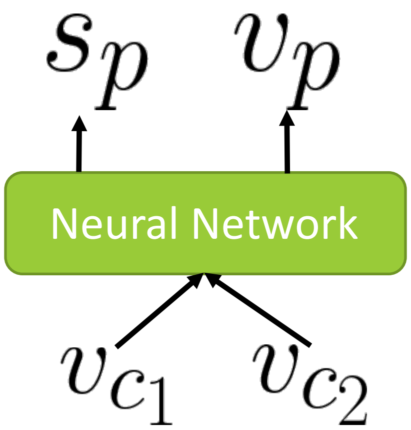
Outline

Property
- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition
- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications
- Parsing
- Paraphrase Detection
- Sentiment Analysis

Recursive Neural Network

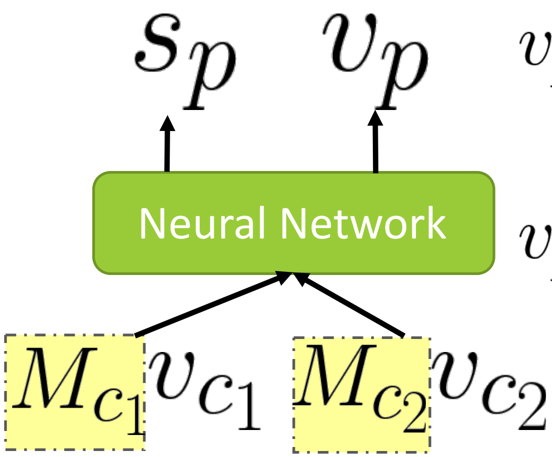


$$v_p = \sigma\left(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b\right)$$

$$s_p = U^T v_p$$

Issue: some words act mostly as an operator, e.g. “very” in “very good”

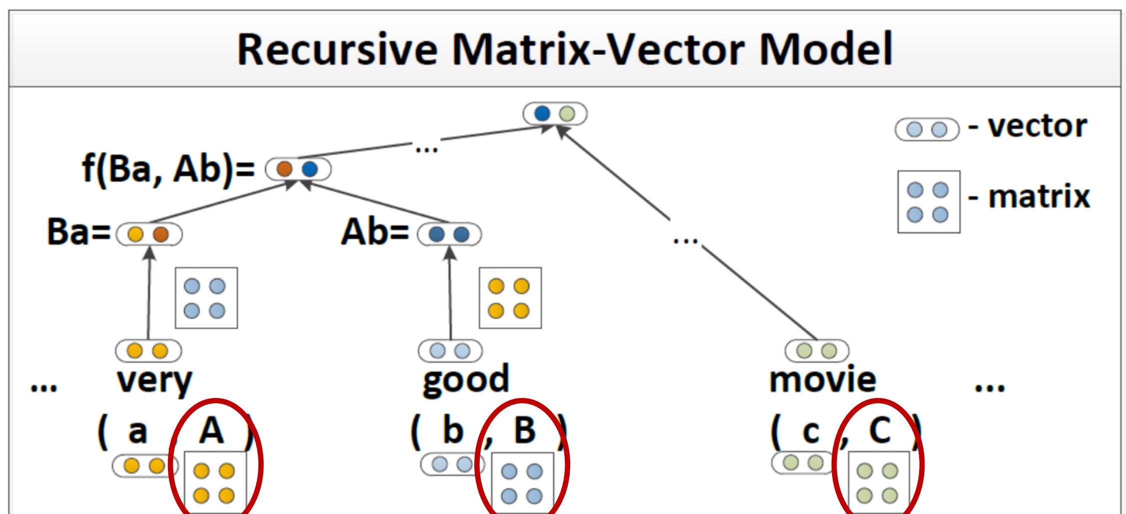
Matrix-Vector Recursive Neural Network



$$v_p = \sigma(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b)$$

$$v_p = \sigma(W \begin{bmatrix} M_{c_2} v_{c_1} \\ M_{c_1} v_{c_2} \end{bmatrix} + b)$$

Idea: each word can additionally serve as an operator



Outline

Property

- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition

- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications

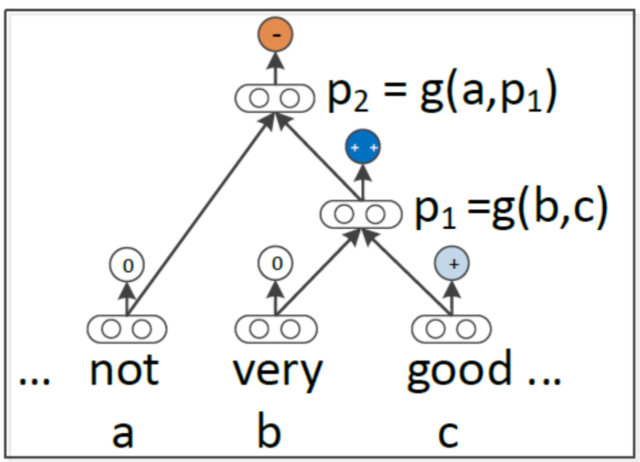
- Parsing
- Paraphrase Detection
- Sentiment Analysis

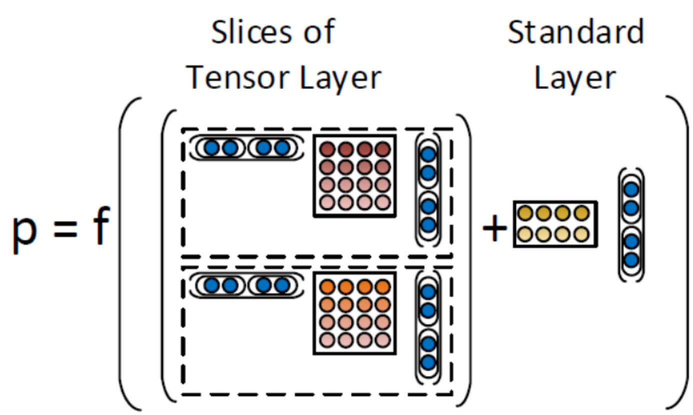
Recursive Neural Tensor Network

$$v_p = \sigma(W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b)$$

Idea: allow more interactions of vectors

$$v_p = \sigma(\begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix}^T V_{c_1,c_2} \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + W \begin{bmatrix} v_{c_1} \\ v_{c_2} \end{bmatrix} + b)$$





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

Outline

Property
- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition
- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications
- **Parsing**
- Paraphrase Detection
- Sentiment Analysis

Language Compositionality

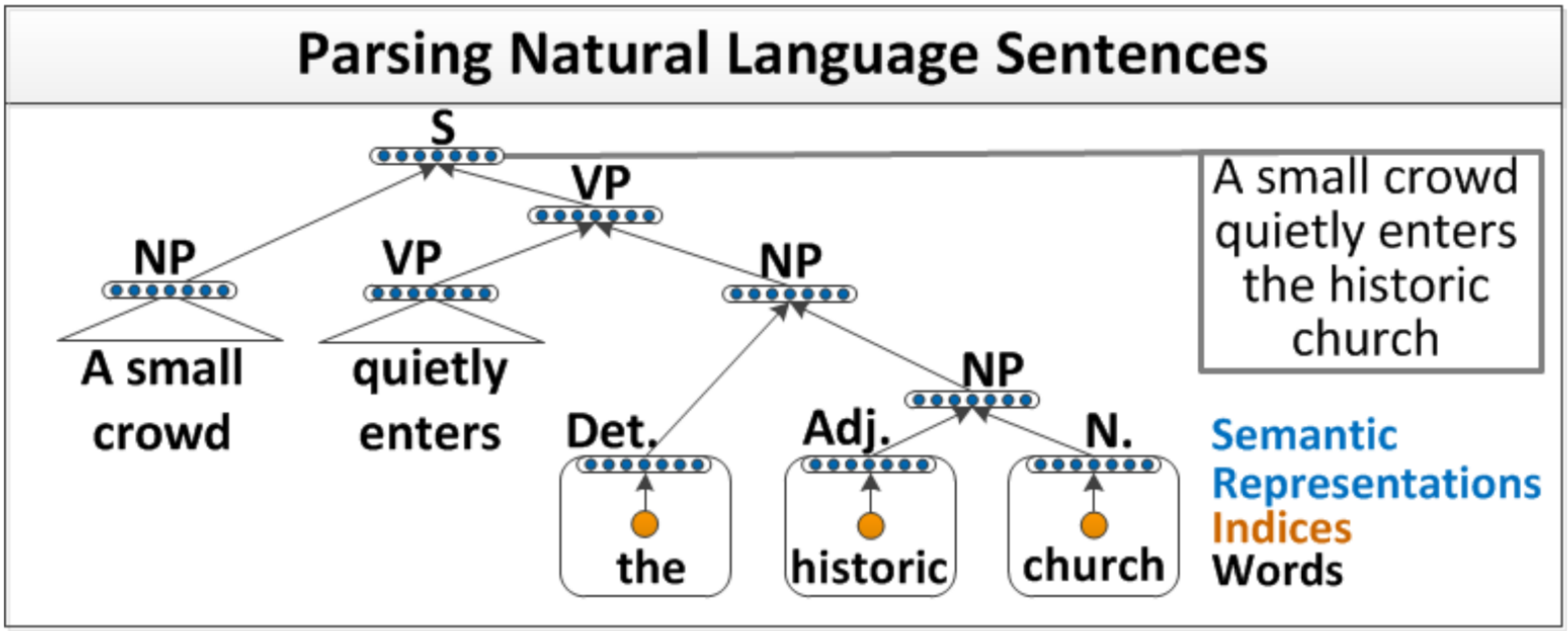
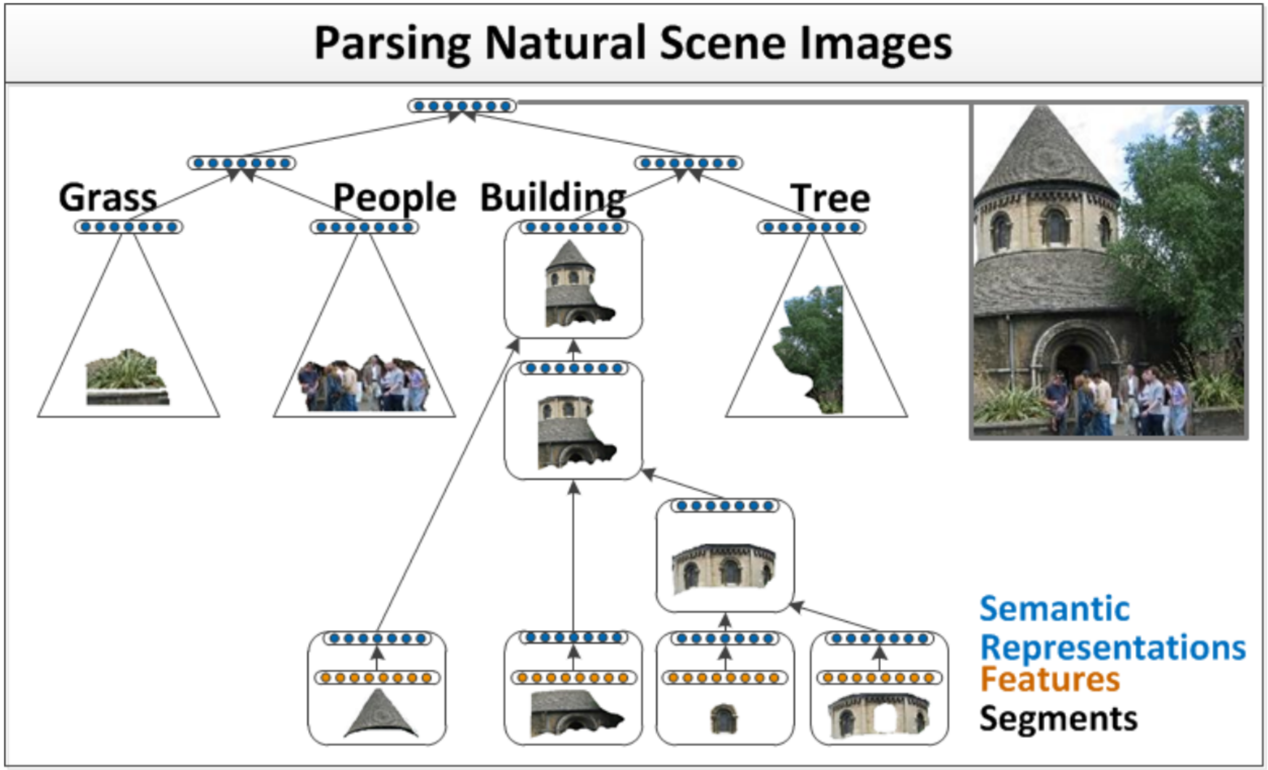


Image Compositionality

Idea: image can be composed by the visual segments (same as natural language parsing)



Outline

Property

- Syntactic Compositionality
- Recursion Assumption

Network Architecture and Definition

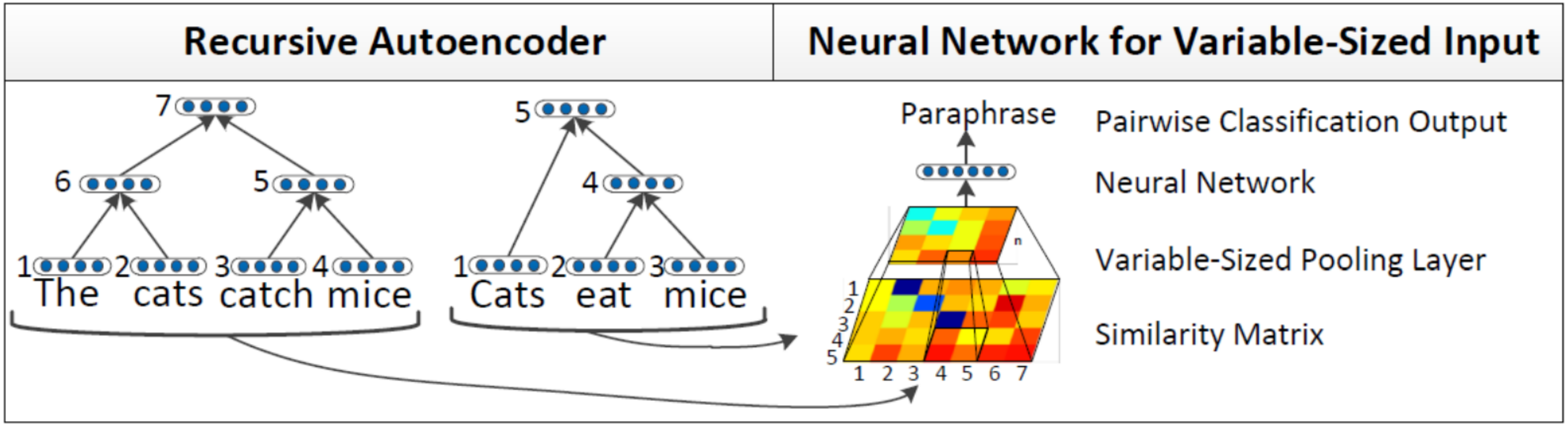
- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications

- Parsing
- **Paraphrase Detection**
- Sentiment Analysis

Paraphrase for Learning Sentence Vectors

A pair-wise sentence comparison of nodes in parsed trees for learning sentence embeddings



Outline

Property

- Syntactic Compositionality
- Recursion Assumption

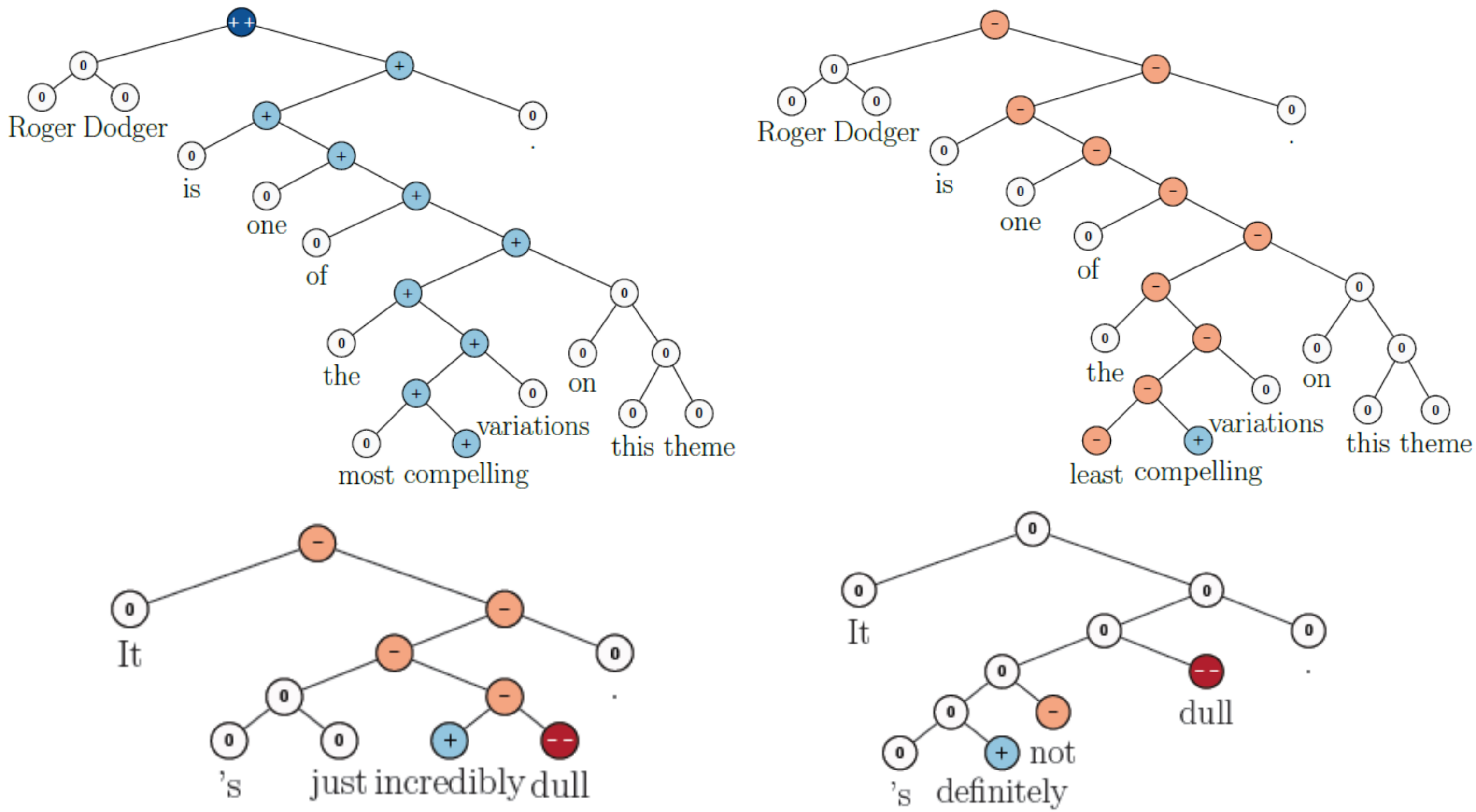
Network Architecture and Definition

- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network

Applications

- Parsing
- Paraphrase Detection
- **Sentiment Analysis**

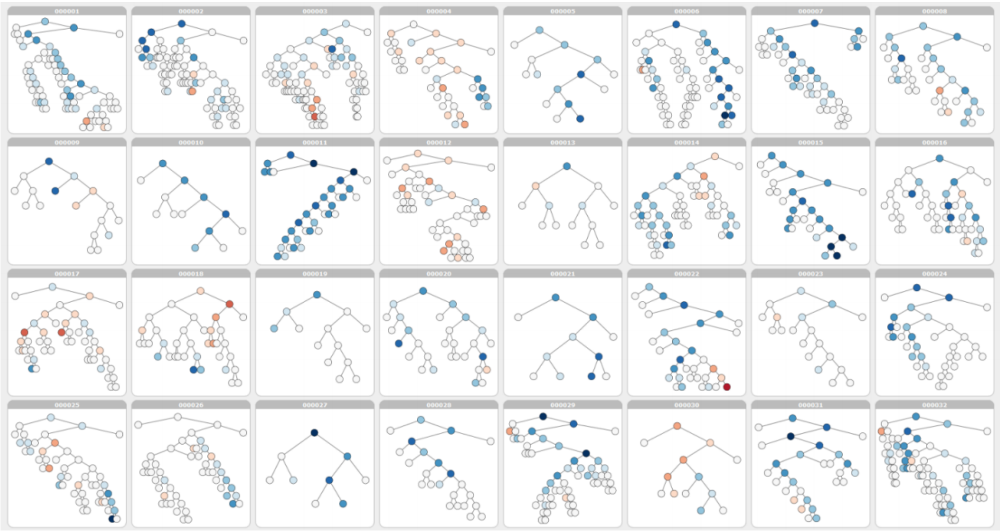
Sentiment Analysis

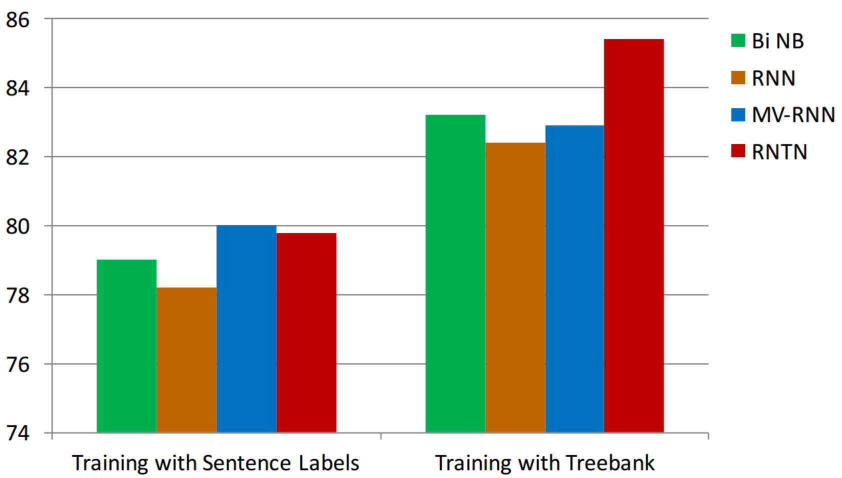


Sentiment analysis for sentences with negation words can benefit from RvNN

Sentiment Analysis

Sentiment Treebank with richer annotations

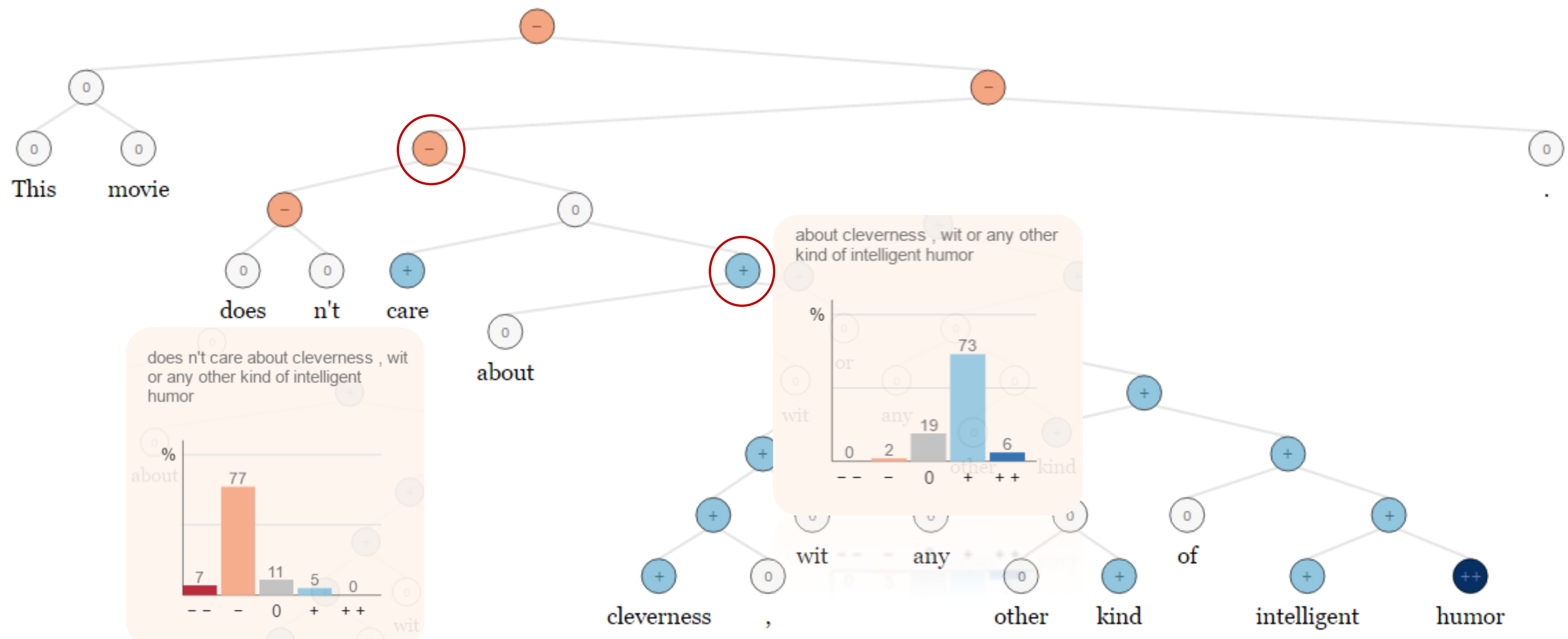




Phrase-level sentiment labels indeed improve the performance

Sentiment Tree Illustration

Stanford live demo: http://nlp.stanford.edu/sentiment/



Phrase-level annotations learn the specific compositional functions for sentiment

Concluding Remarks

Recursive Neural Network

- Idea: syntactic compositionality & language recursion

Network Variants

- Standard Recursive Neural Network
  - Weight-Tied
  - Weight-Untied
- Matrix-Vector Recursive Neural Network
- Recursive Neural Tensor Network