Recursive Neural Network (RvNN)

WeiYang
51184506043
weiyang@godweiyang.com
https://godweiyang.com

Outline

- Introduction
- Structure of RvNN
- Backpropagation through structure (BPTS)
- More complex variants
- Applications
- Project

Compositionality

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

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

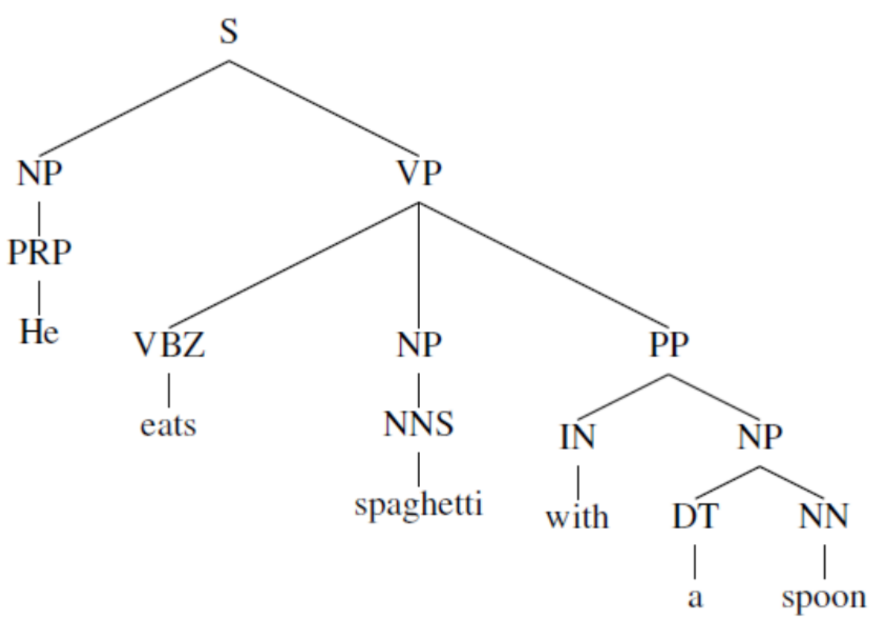
People interpret the meaning of larger text units entities, descriptive terms, facts, arguments, stories by semantic composition of smaller elements

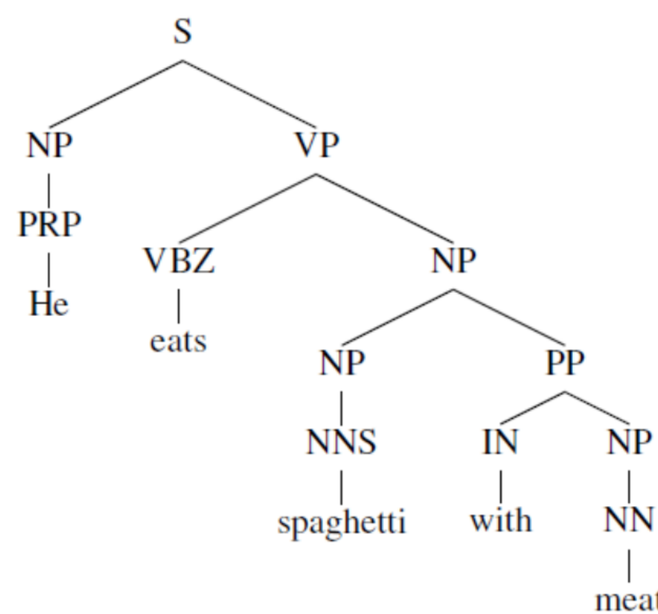
Recursion

- Cognitively somewhat debatable
- But recursion is natural for describing language
  - [The man from [the company that you spoke with about [the project] yesterday]]
- Noun phrase containing a noun phrase containing a noun phrase

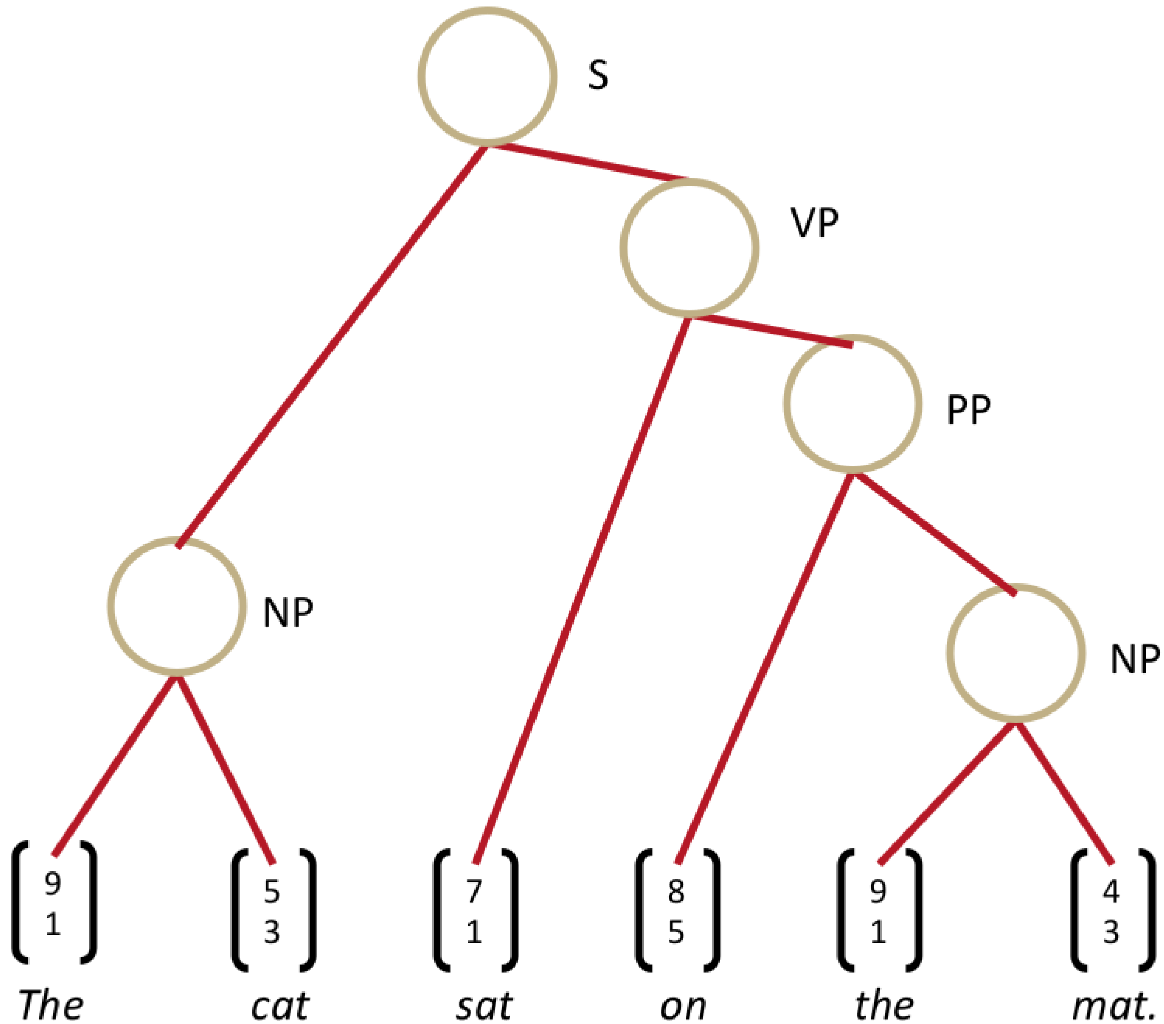
Ambiguation

RvNN is helpful in disambiguation. However, recurrent neural network (RNN) can't do it.

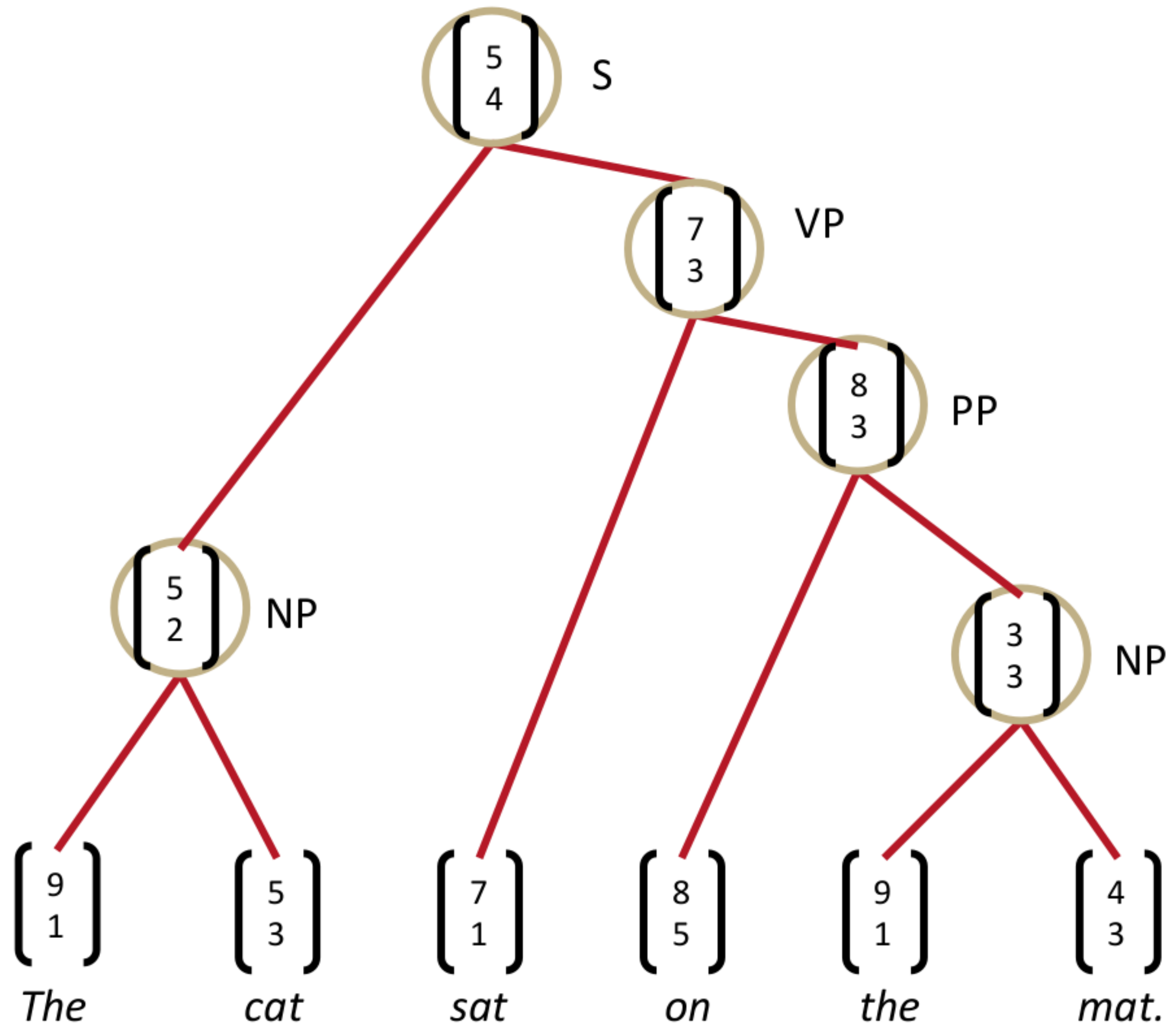




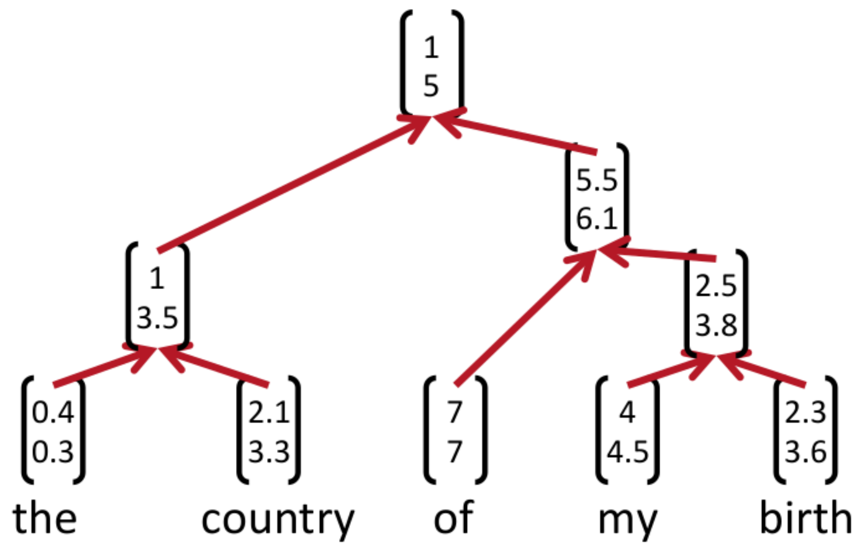
Constituency Tree

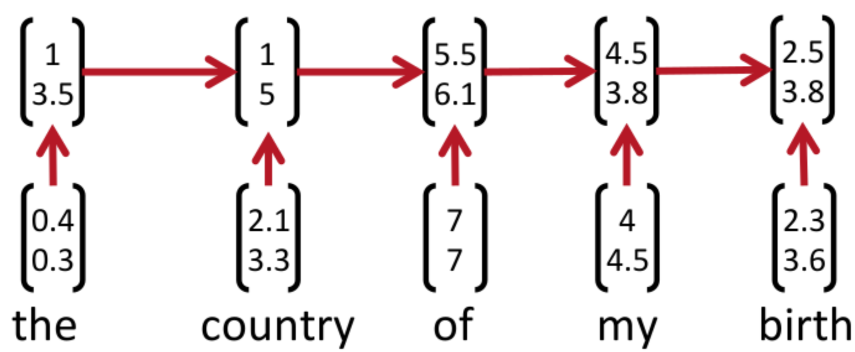


Learn Structure and Representation



RvNN vs RNN

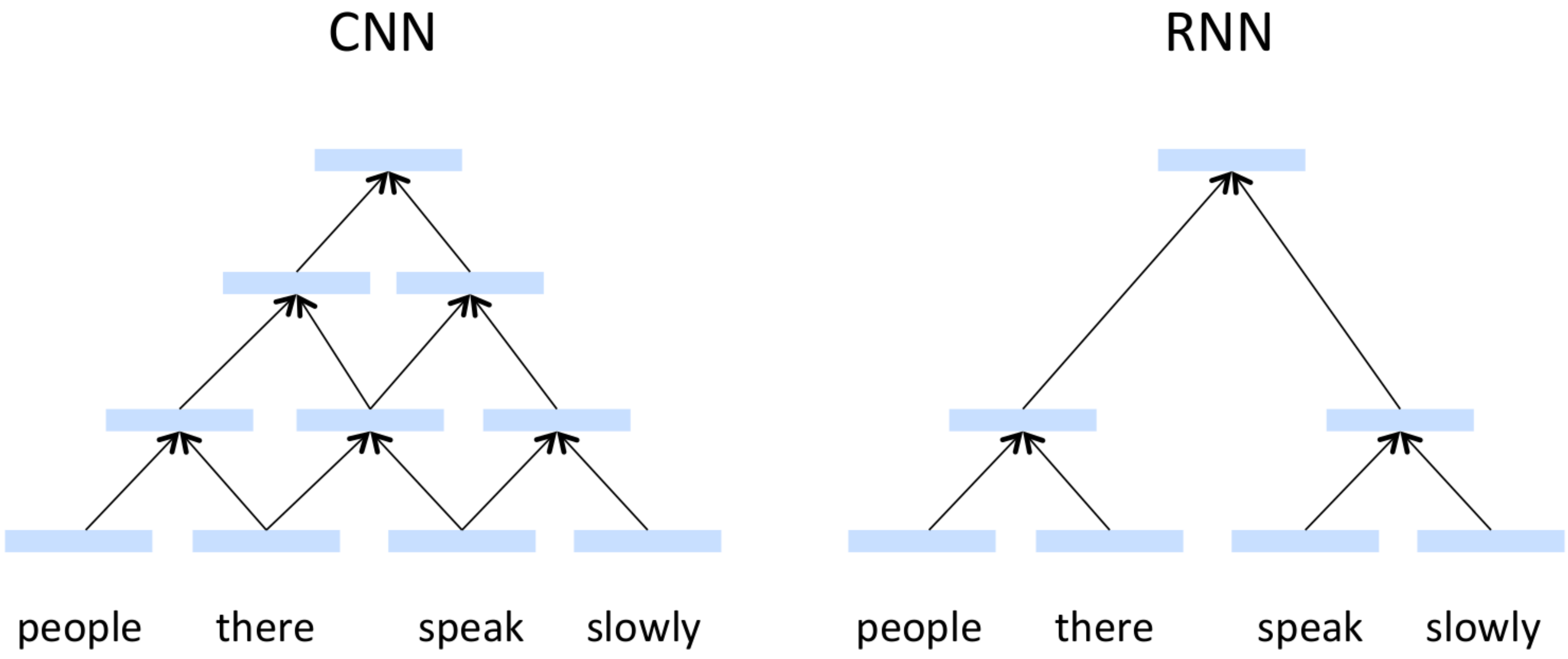

1
5
5.5
6.1
1
3.5
2.5
3.8
0.4
0.3
2.1
3.3
7
7
4
4.5
2.3
3.6
the
country
of
my
birth


1
3.5
1
5
5.5
6.1
4.5
3.8
2.5
3.8
0.4
0.3
2.1
3.3
7
7
4
4.5
2.3
3.6
the
country
of
my
birth

RvNN vs RNN

- Recursive neural nets require a parser to get tree structure
- Recurrent neural nets cannot capture phrases without prefix context and often capture too much of last words in final vector

RvNN vs CNN

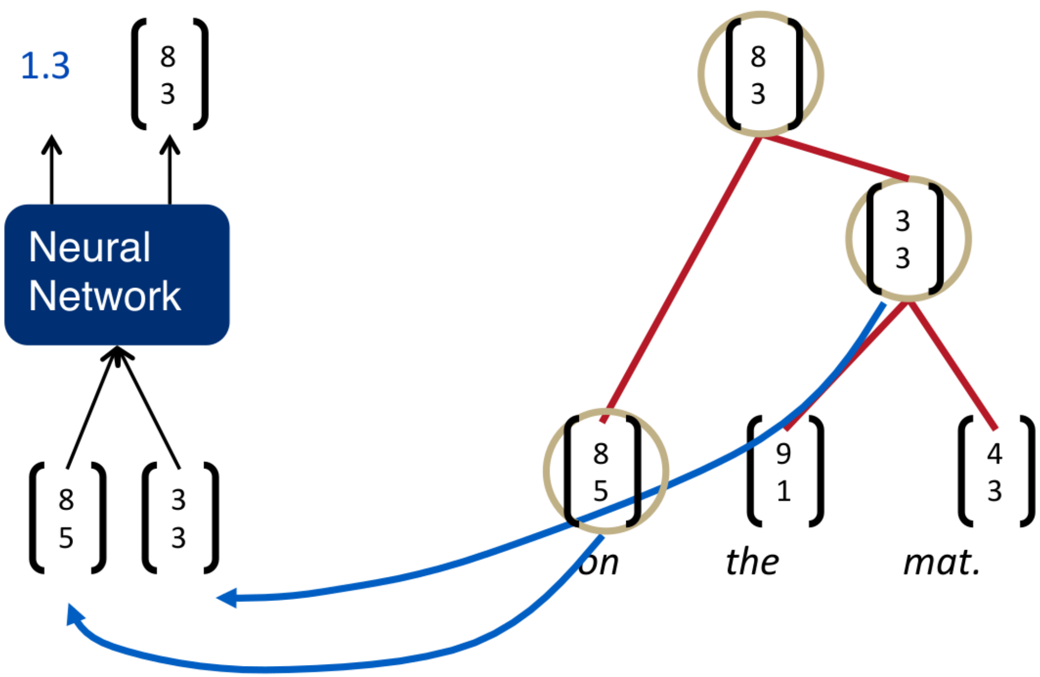

CNN
people
there
speak
slowly
RNN
people
there
speak
slowly

RvNN vs CNN

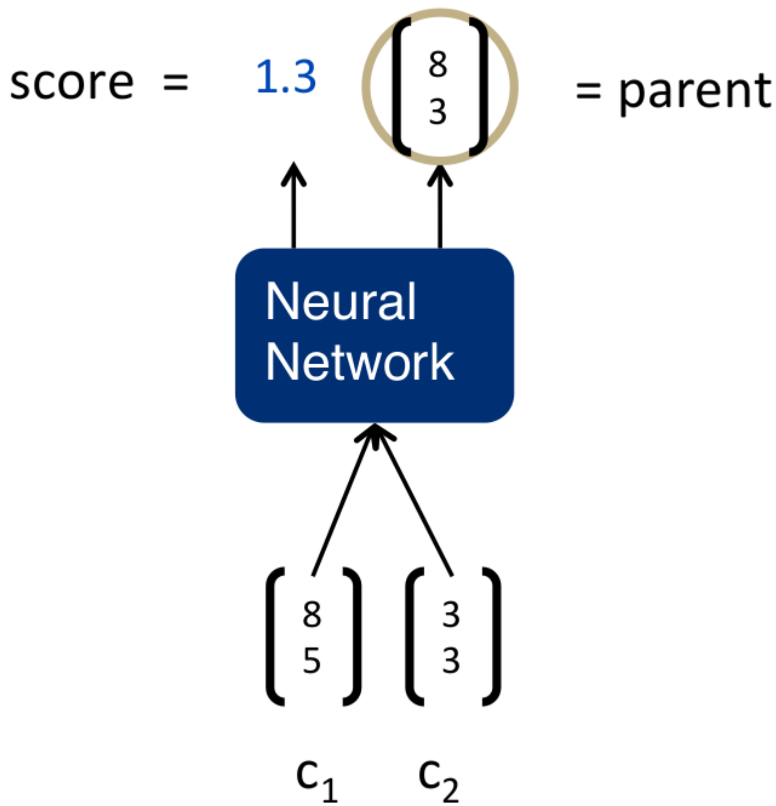
- RvNN get compositional vectors for grammatical phrases only
- CNN computes vectors for every possible phrase
    - Regardless of whether each is grammatical and many don't make sense
    - Don't need parser
    - But maybe not very linguistically or cognitively plausible

RvNN for Structure Prediction

- Inputs: two candidate children's representations
- Outputs:
  - The semantic representation if the two nodes are merged
  - Score of how plausible the new node would be



RvNN Definition

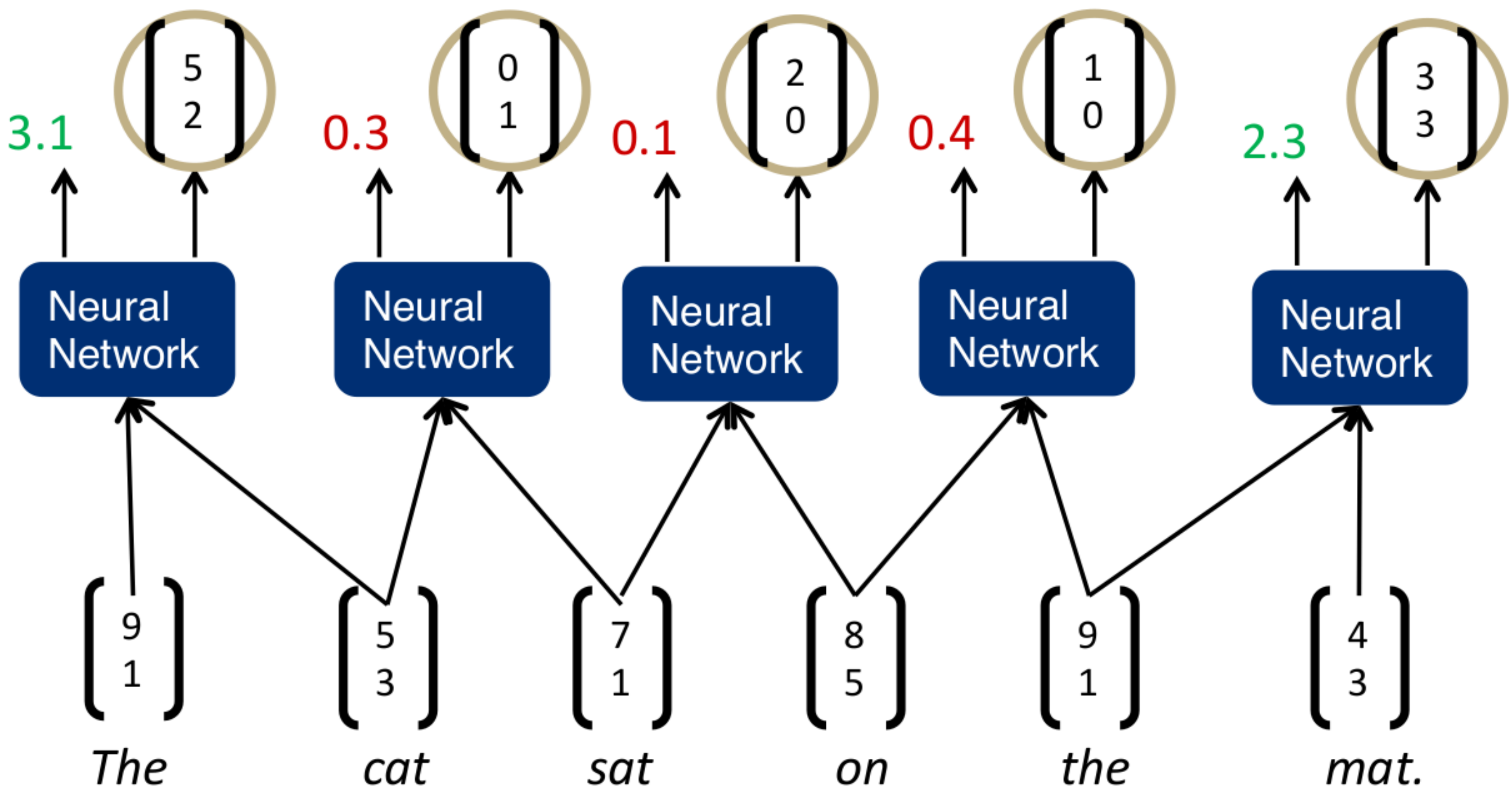


score $= U^{\mathsf{T}} p$

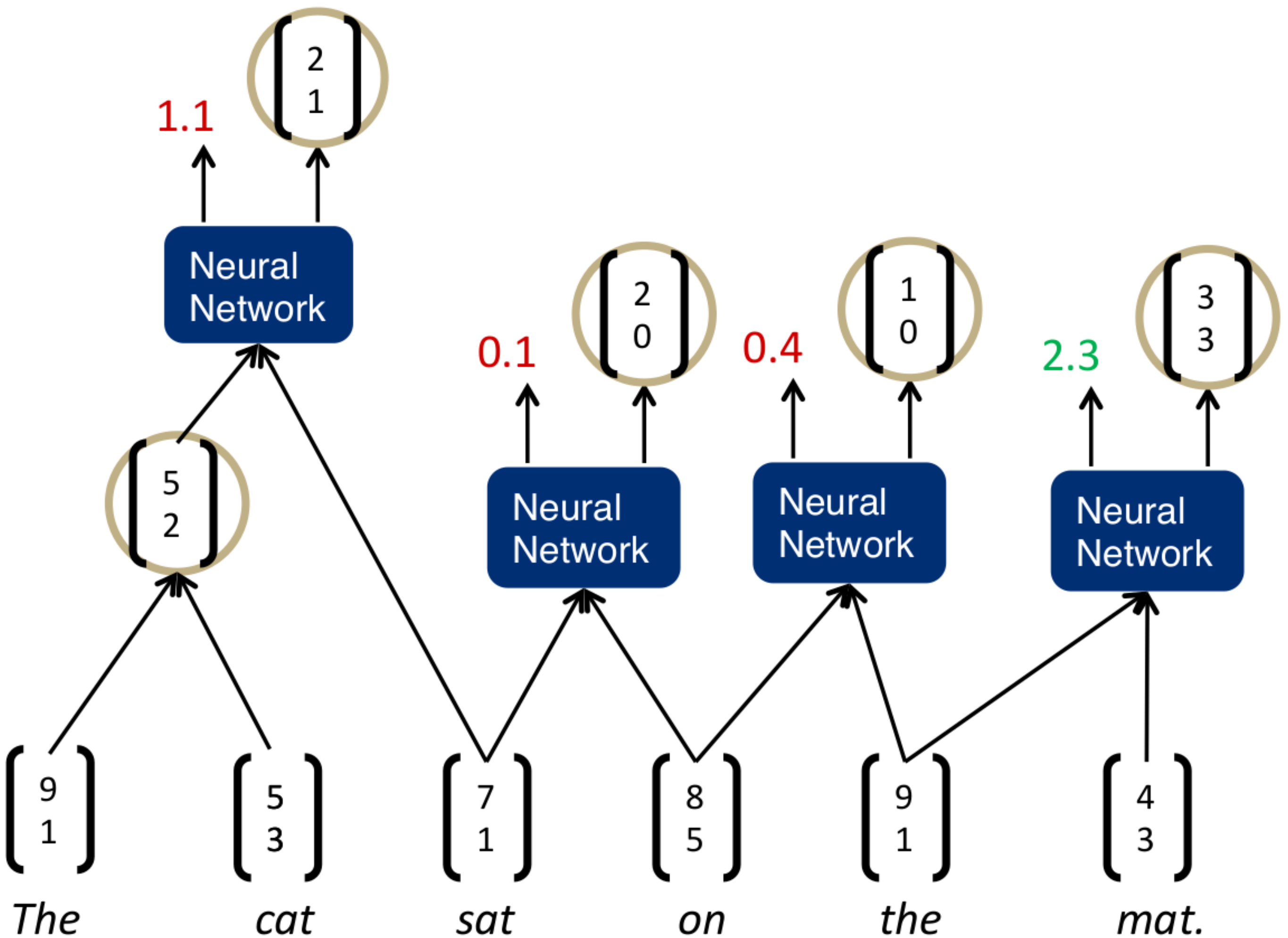
$$\boxed{p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),}$$

Same *W* parameters at all nodes of the tree

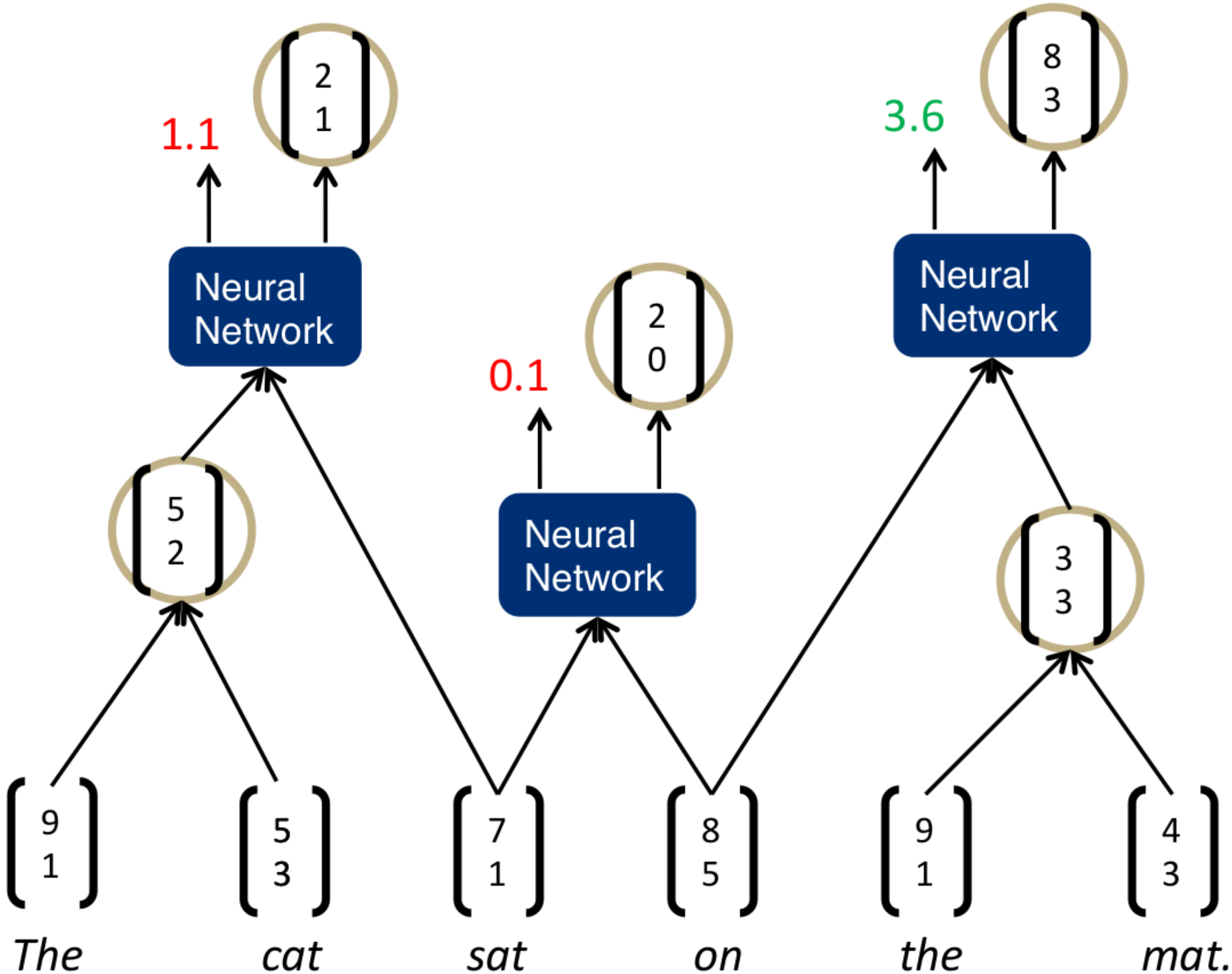
Parsing a sentence with RvNN



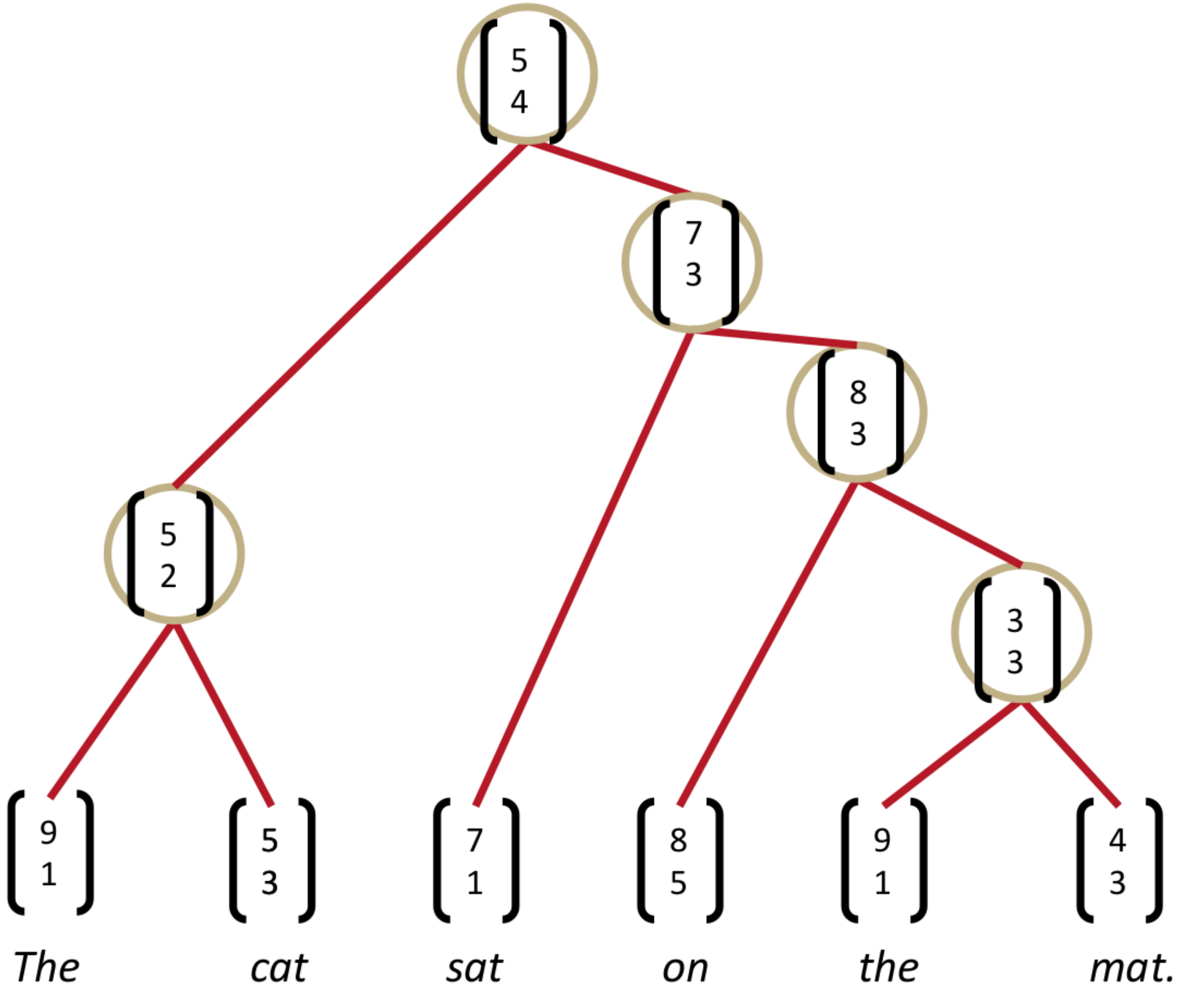
Parsing a sentence with RvNN



Parsing a sentence with RvNN



Parsing a sentence with RvNN



Max-Margin Framework

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

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

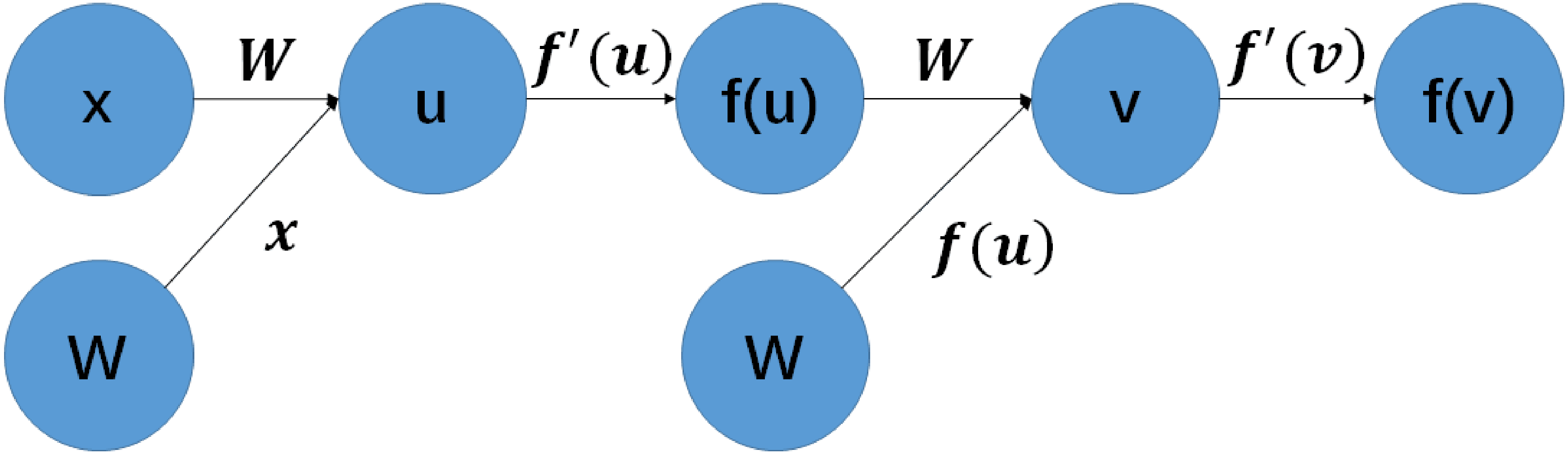
- Supervised max-margin objective:

$$\mathcal{L}(\theta) = \max(0, s(x_i, \hat{y}) + \Delta(y_i, \hat{y}) - s(x_i, y_i))$$

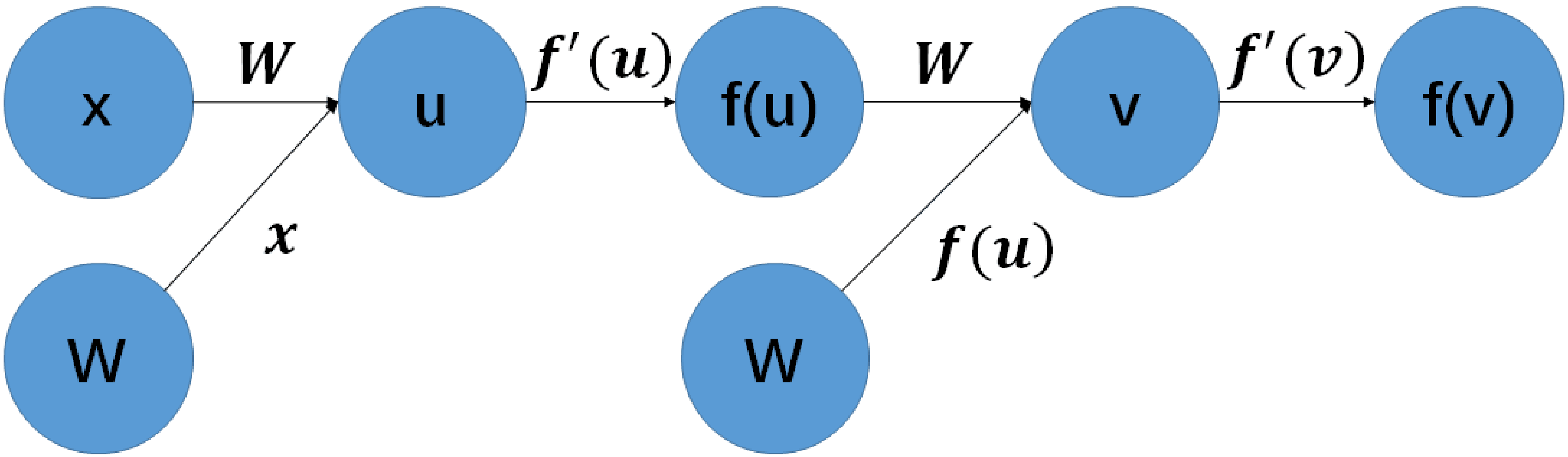
- Structure search for $\hat{y}$ can use greedy, beam search and dynamic programming

Computational Graph

Question: How can we compute $\frac{\partial}{\partial W} f(W(f(Wx)))$?



Computational Graph



$$\frac{\partial}{\partial W} f(W(f(Wx)))$$

$$= f'(v)Wf'(u)x + f'(v)f(u)$$

$$= f'(v)(Wf'(u)x + f(u))$$

$$= f'(Wf(Wx))(Wf'(Wx)x + f(Wx))$$

Backpropagation Through Structure

Principally the same as general backpropagation

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

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

Three differences resulting from the recursion and tree structure:

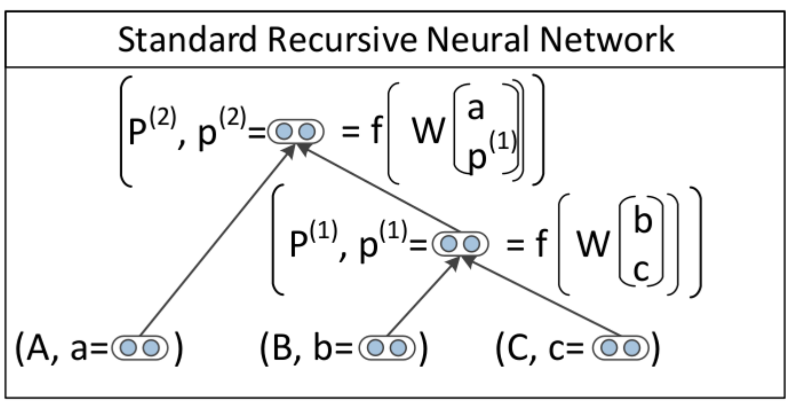
- Sum derivatives of W from all nodes (like RNN)
- Split derivatives at each mode (for tree)
- Add error messages from parent + node itself

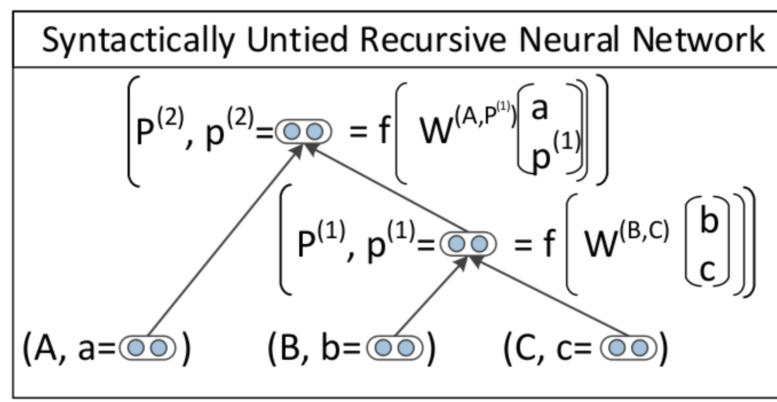
Problems with Simple RvNN

- Single weight matrix RvNN could capture some phenomena but not adequate for more complex, higher order composition and parsing long sentences
- There is no real interaction between the input words
- The composition function is the same for all syntactic categories, punctuation, etc.
- Gradient vanishing

Syntactically-Untied RvNN

- 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
- A TreeRNN can do better with different composition matrix for different syntactic environments
- The result gives us a better semantics

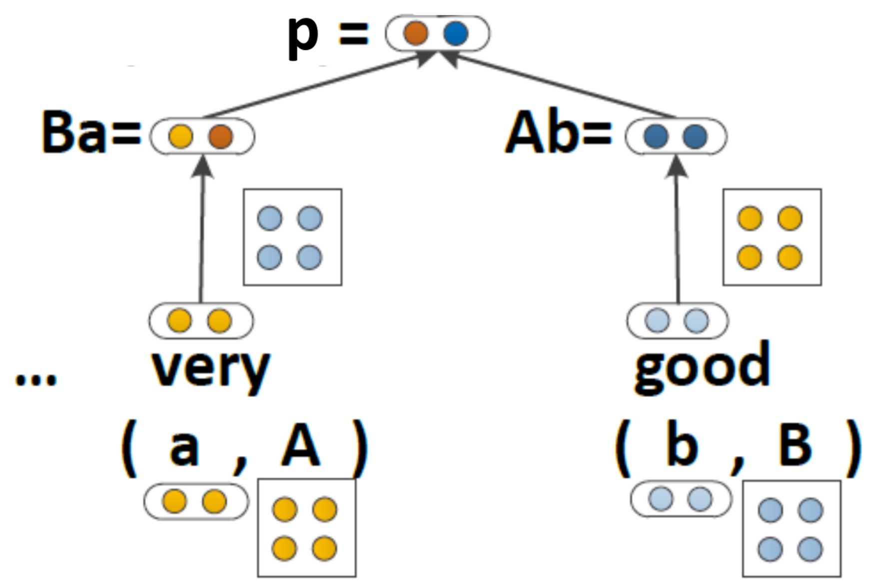




Matrix-Vector RvNN

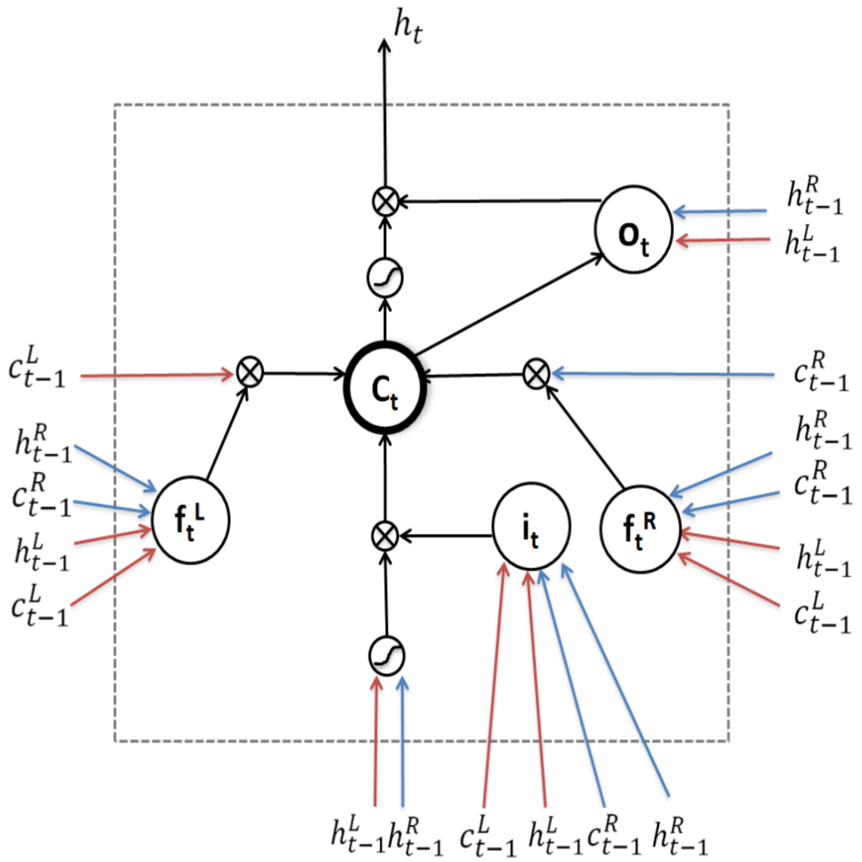
- Some words act mostly as an operator, e.g. "very" in "very good"
- MV-RvNN

$$p = f(W[Ba; Ab]^T + b)$$



Tree LSTM

- Avoid gradient vanishing and can model long-distance interaction over trees



Compositional Vector Grammars

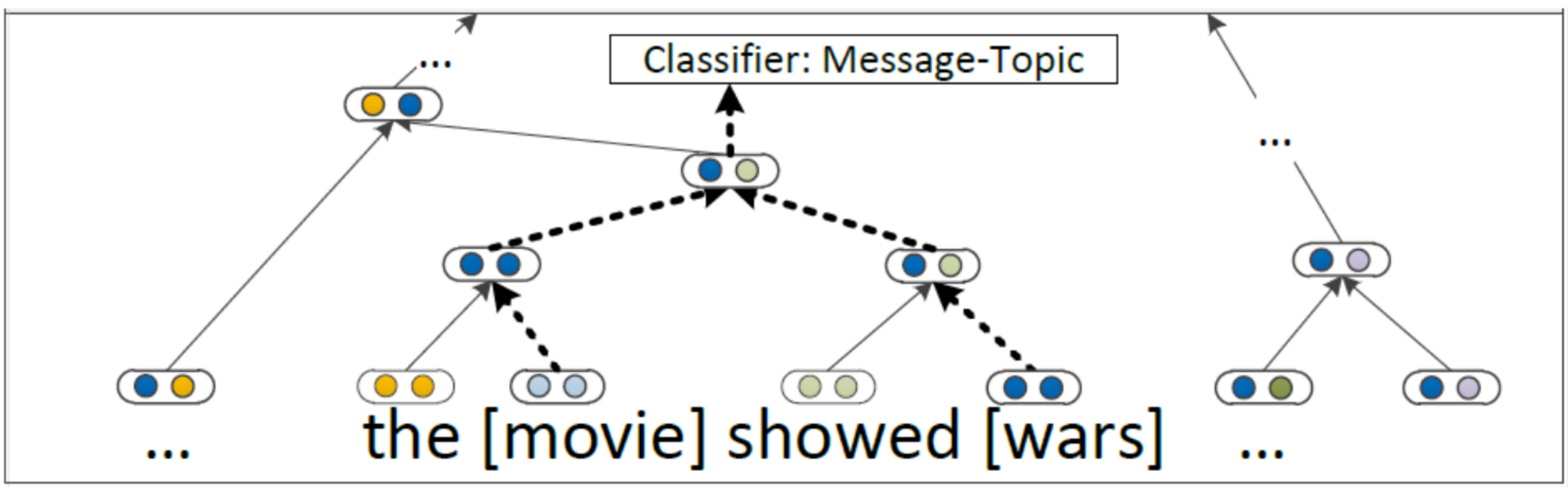
- CVG = PCFG + RvNN
- Scores at each node computed by combination of PCFG and SU-RNN:

$$s(p^{(1)}) = (v^{(B,C)})^T p^{(1)} + \log P(P_1 \to BC)$$

- **Socher et al. ACL 2013**

Semantic Representations

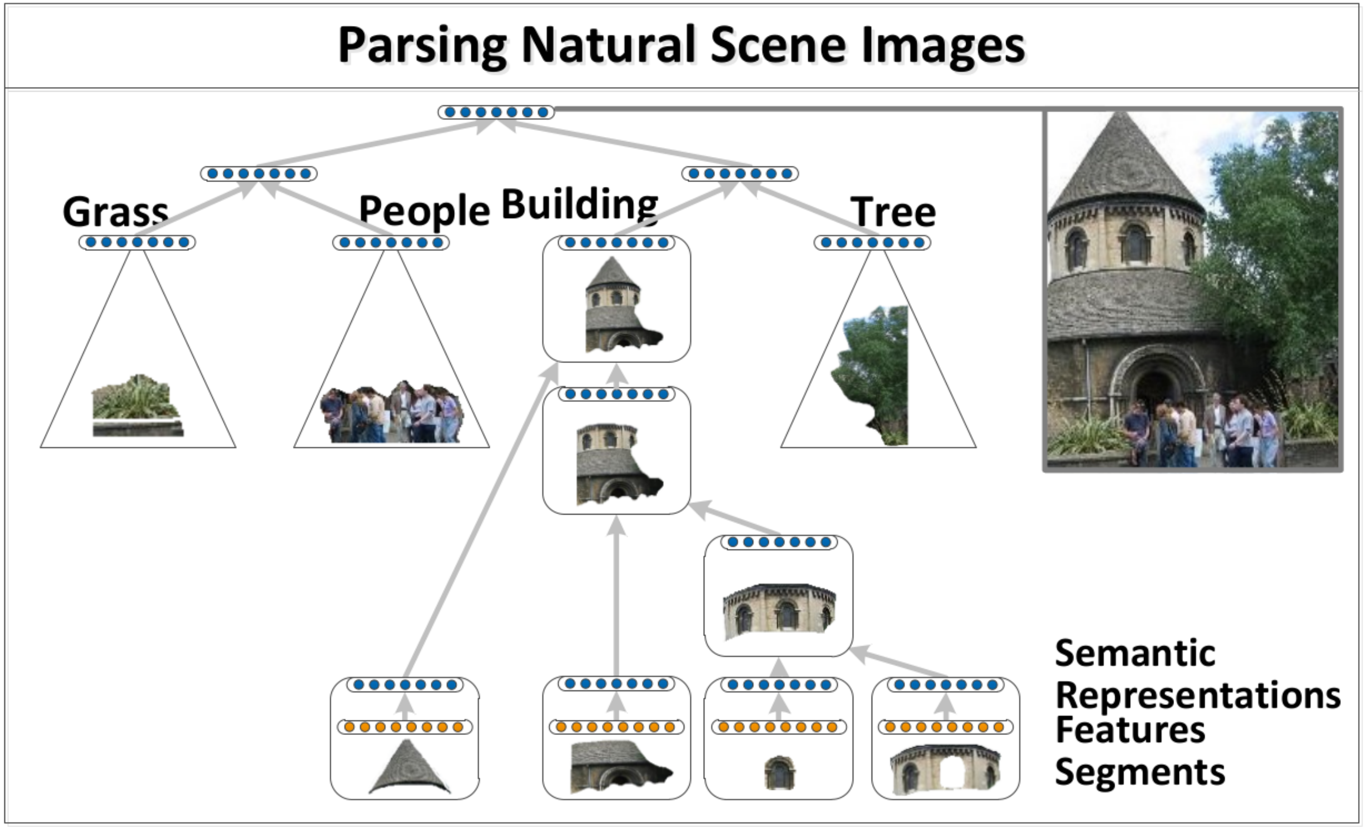
- **Semantic Relatedness:** Build a single compositional semantics for the minimal constituent including both terms



- **Sentiment Classification**

Scene Parsing

- Same Recursive Neural Network as for natural language parsing **(Socher et al. ICML 2011)**



Project

- **Data Format (PTB)**
  (SBARQ (**WHADVP When**) (SBARQ (SQ (**VERB did**) (SQ (**NP Nixon**) (**VP die**))) (. ?)))
- **Labeled Precision (LP)**
  $LP = (True\ predicted\ spans)/(Total\ predicted\ spans)$
- **Labeled Recall (LR)**
  $LR = (True\ predicted\ spans)/(Total\ gold\ spans)$
- **F1**
  $F1 = (2 \times LP \times LR)/(LP + LR)$

Project

- **Deep Learning Framework**

  Dynet (recommended), Tensorflow, Pytorch, Keras, etc.

- **Grading**
  - Submission on time. (50')
  - Code. (20')
  - Results. (10')
  - Report. (20')

References

A Summary of Constituent Parsing
https://github.com/godweiyang/ConstituentParsing