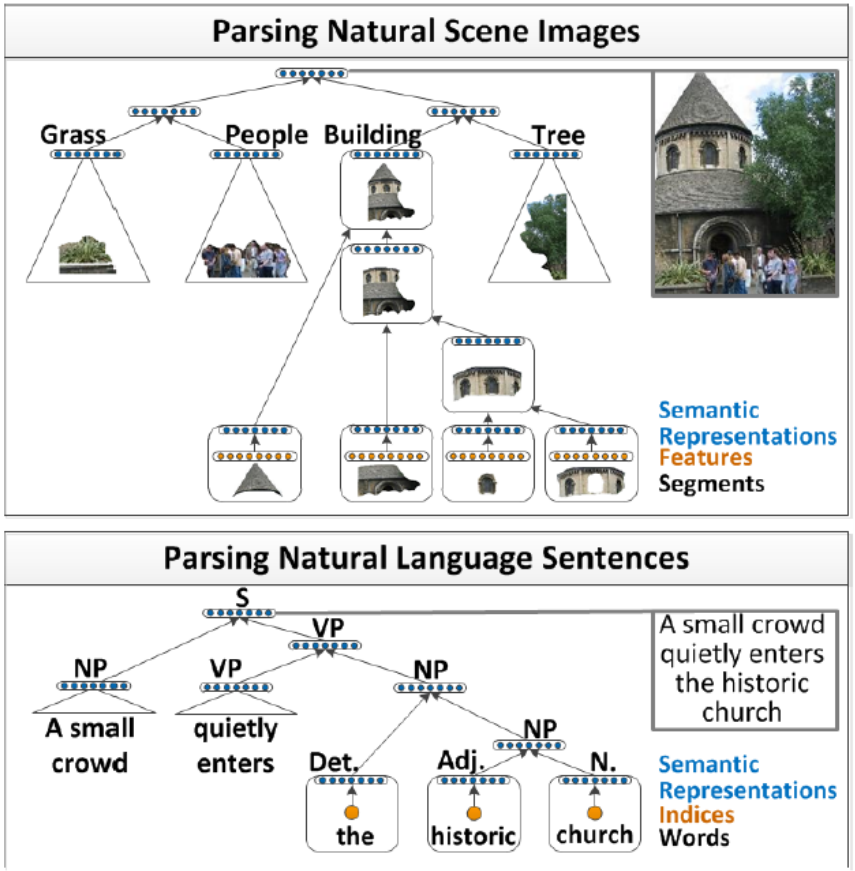
分布式语义组合
Distributed Semantic Composition

车万翔
社会计算与信息检索研究中心
2017年春季学期

Semantic Compositionality

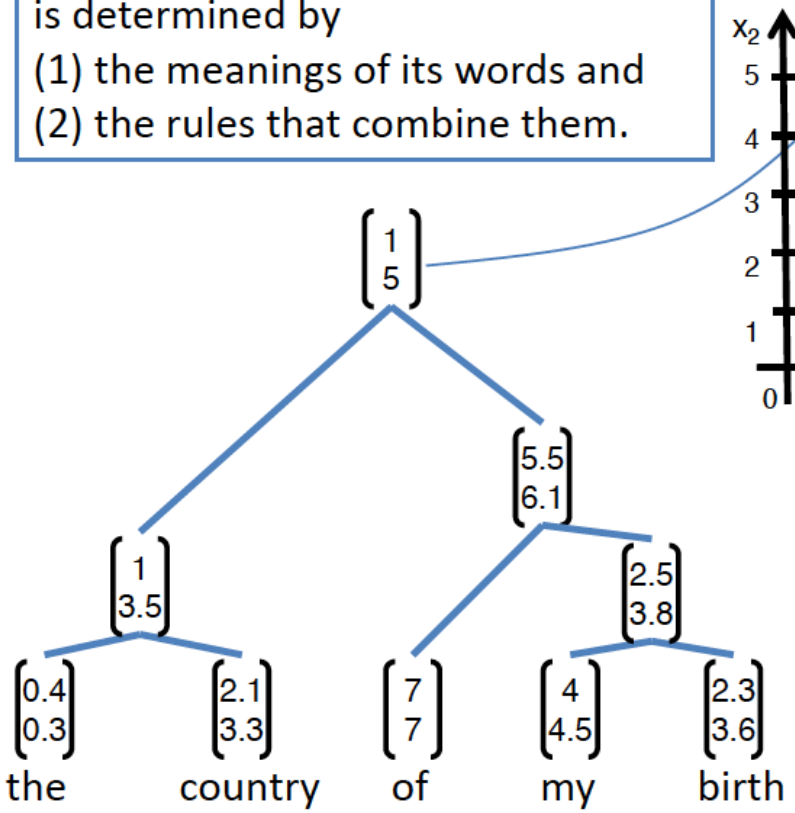
- Meaning representations can be composed from smaller units
- Useful for many different tasks in NLP (and vision)

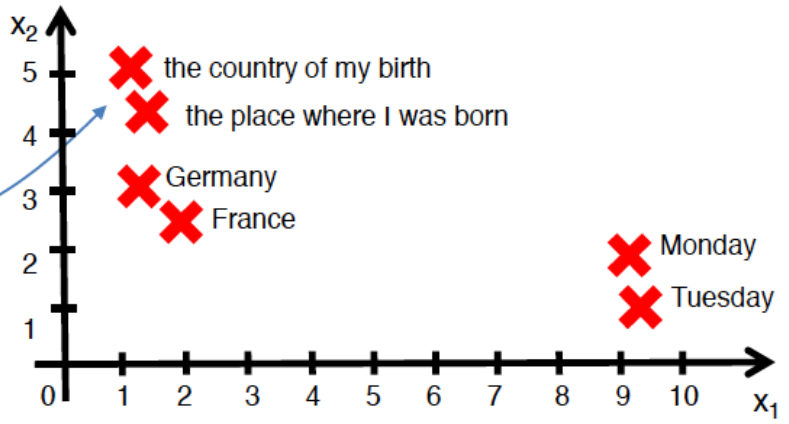


Represent Phrases as Vectors

Use the 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.

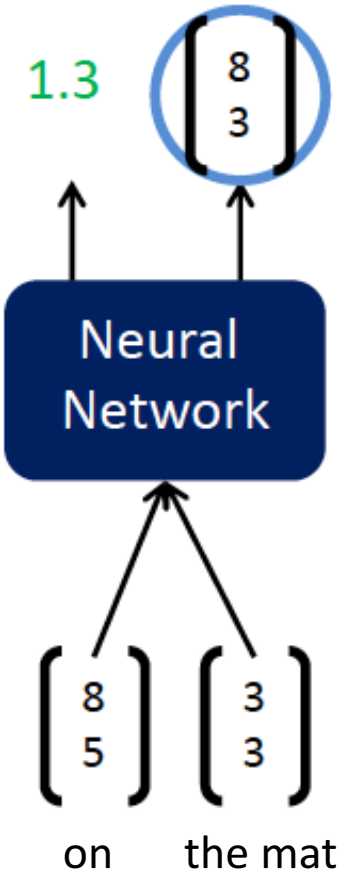




Model jointly learns compositional vector representations and tree structure.

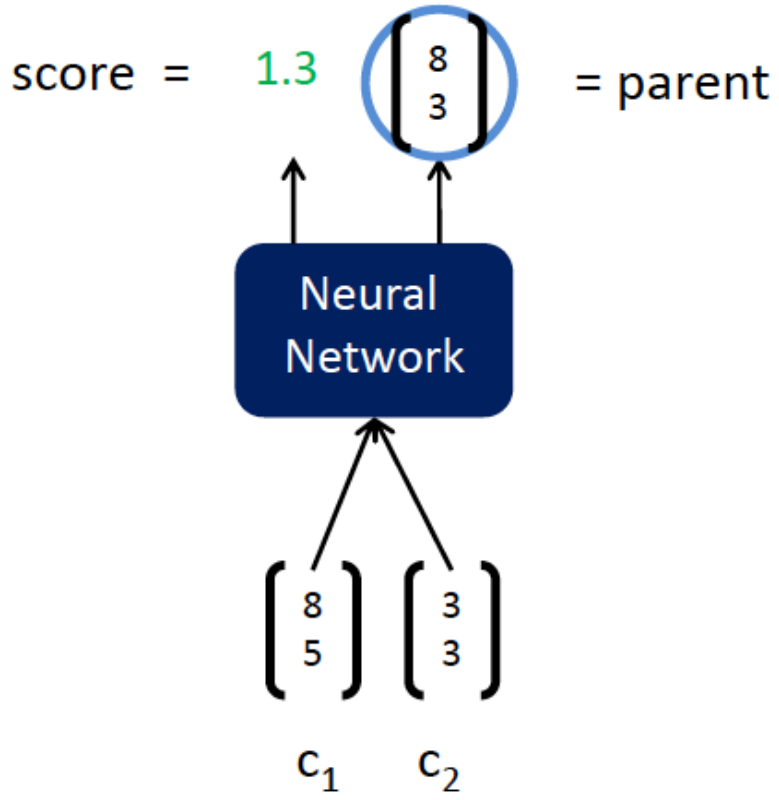
Recursive NN for Phrase Vectors

- 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



Recursive NN Definition

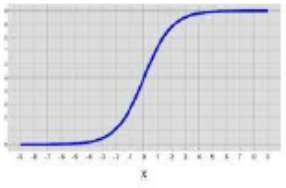
- Socher et al. (ICML, 2011)



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

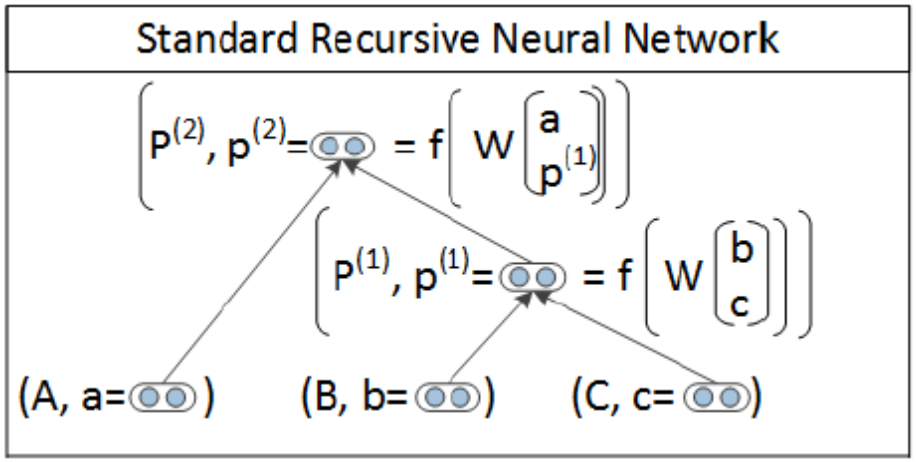
$$p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),$$

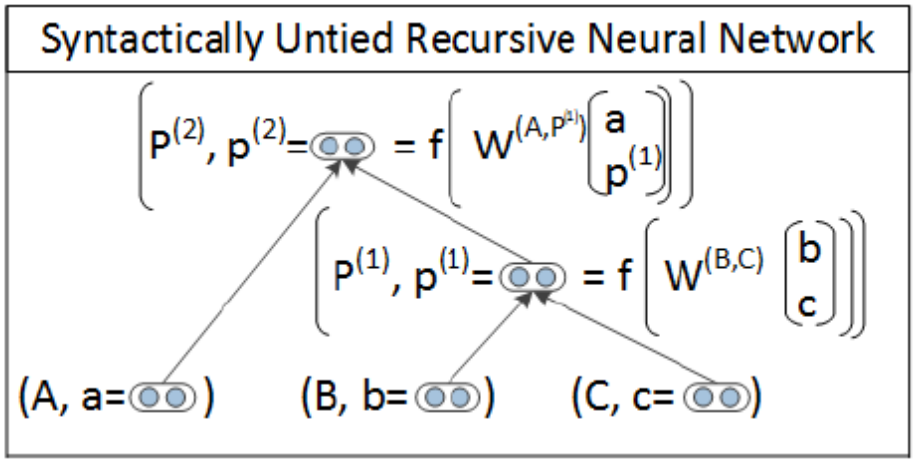
where tanh:



Syntactically Untied RNN

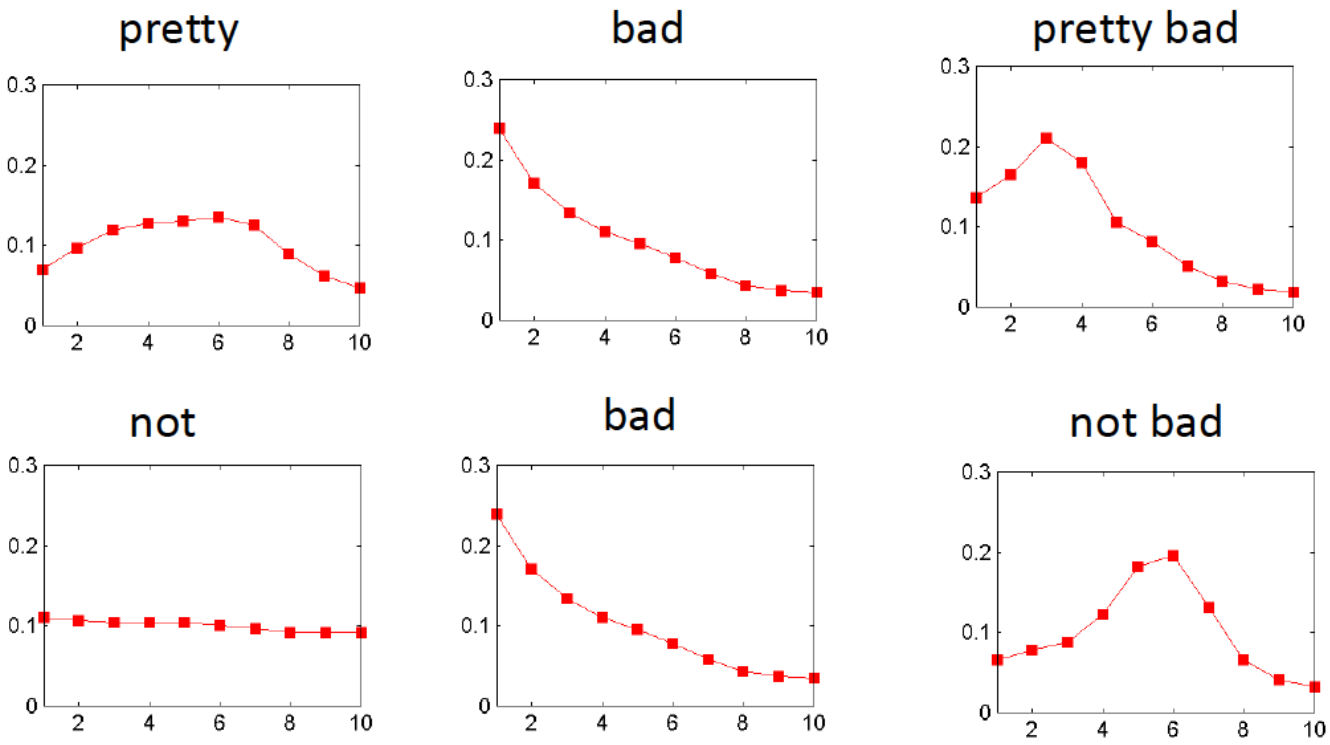
- Use a PCFG as a syntactic backbone to prune unlikely candidates
- Provide discrete syntactic categories on which we can condition the continuous composition function





Limitations of Matrix Composition

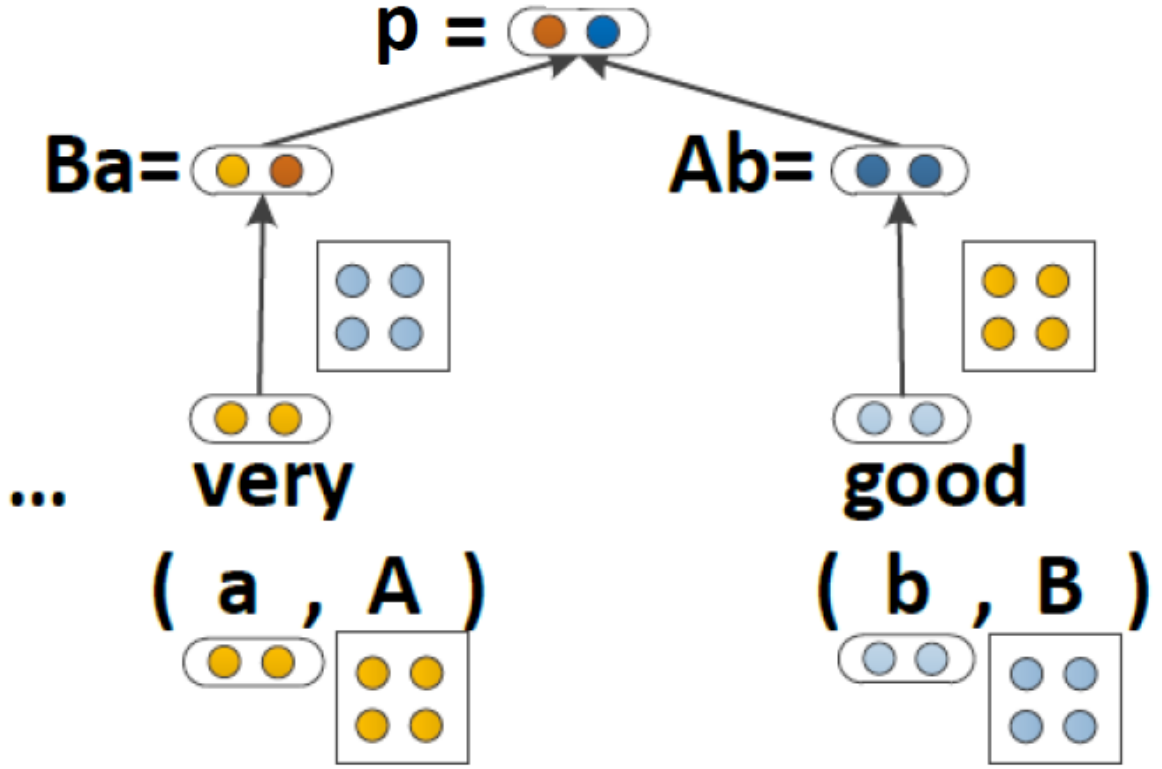
- Cannot capture word-specific operational meaning
- Some semantic language phenomena require richer compositional functions or operator representations for words



Matrix-vector Representations

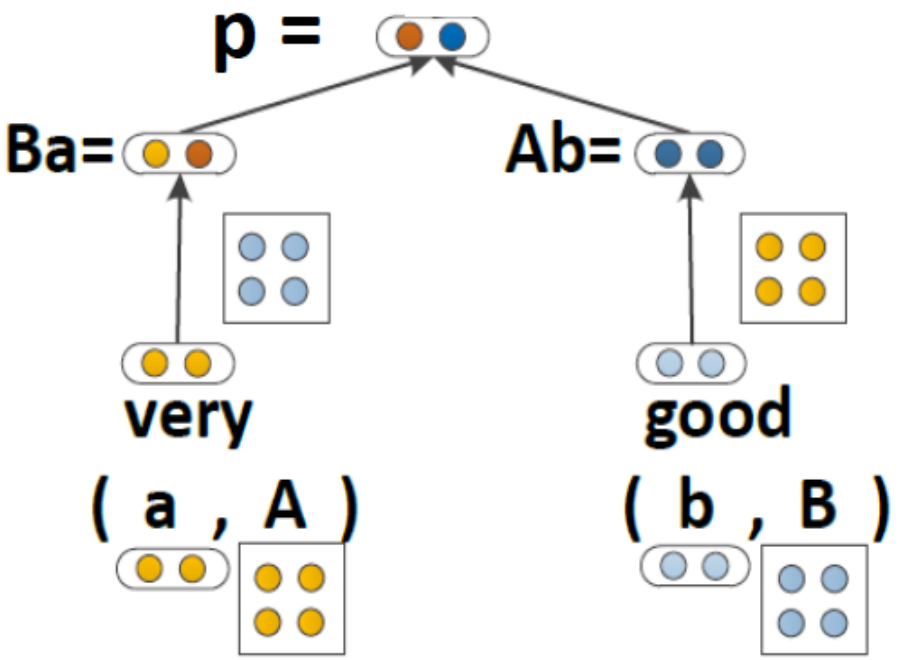
$$p = f\left(W \begin{bmatrix} a \\ b \end{bmatrix}\right)$$

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



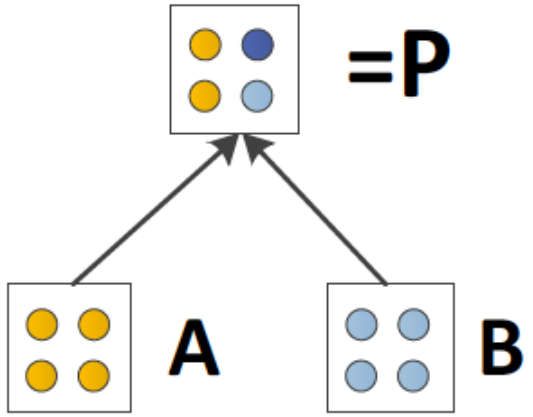
Computing New Matrix

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

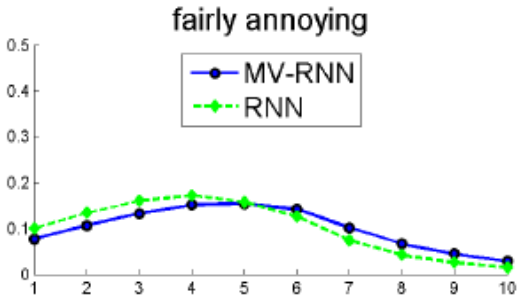


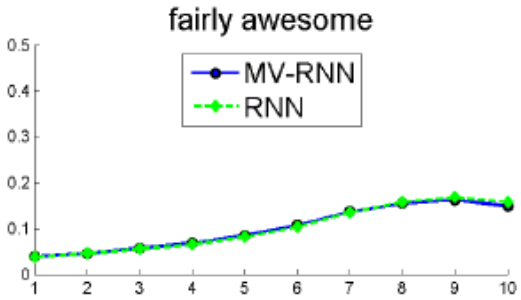
$$P = g(A, B) = W_M \begin{bmatrix} A \\ B \end{bmatrix}$$

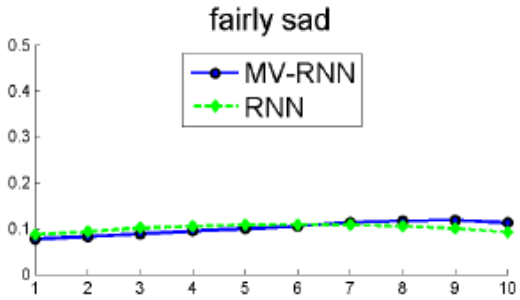
$$W_M \in \mathbb{R}^{n \times 2n}$$

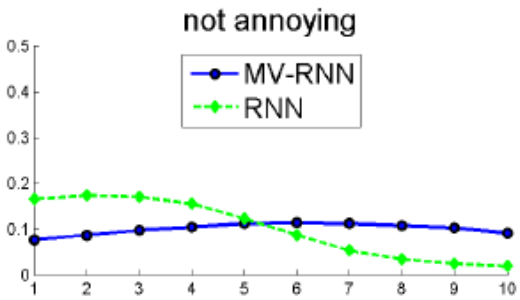


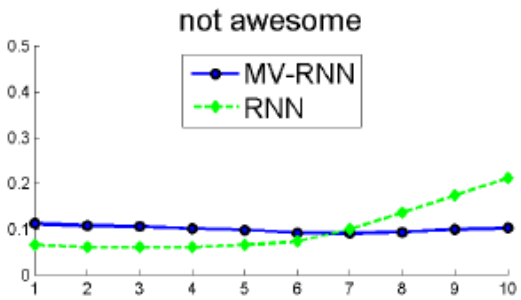
Examples

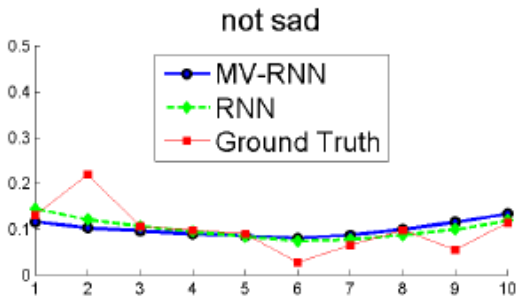

fairly annoying
MV-RNN
RNN

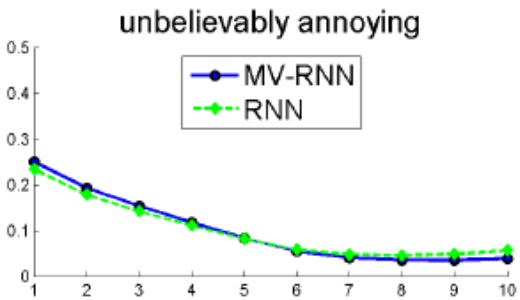

fairly awesome
MV-RNN
RNN

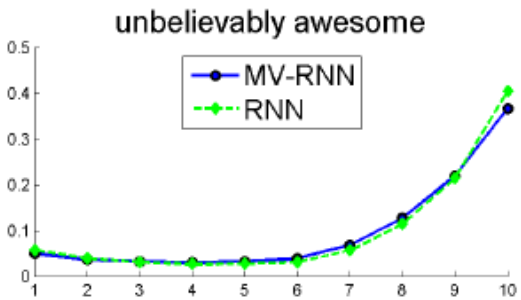

fairly sad
MV-RNN
RNN

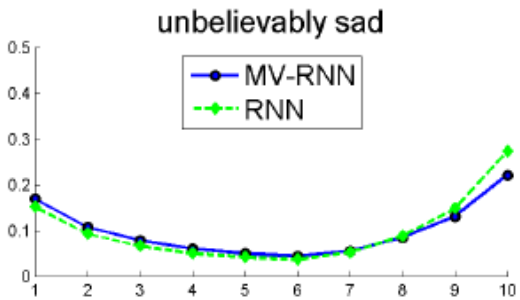

not annoying
MV-RNN
RNN


not awesome
MV-RNN
RNN


not sad
MV-RNN
RNN
Ground Truth


unbelievably annoying
MV-RNN
RNN


unbelievably awesome
MV-RNN
RNN


unbelievably sad
MV-RNN
RNN

Recursive Neural Tensor Networks

- An Application
  - Sentiment Analysis (also known as opinion mining)
  - Aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document

61 of 61 people found the following review helpful

★★★★★ **Great Beginner's Book**

By AS on February 24, 2011

Format: Paperback | **Verified Purchase**

I was new to Python and fairly new to programming when I read this book. This book is extremely clear and well-written. It introduces a novice to the foundational concepts of computer science. There are many great examples and activities that the reader can jump into almost immediately. I had already written my first Python program after 10 minutes of reading.

I highly recommend this book for beginners.

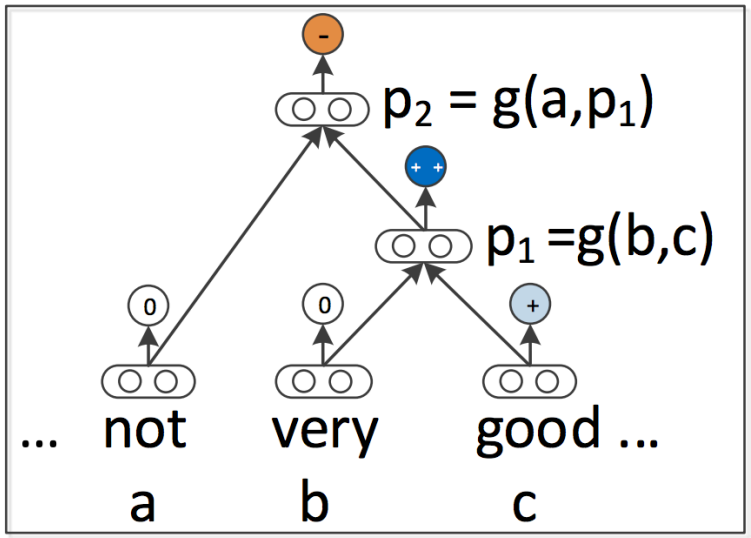
Sentiment Detection and BOW Models

- Detection accuracy for longer documents ~90%
  - Lots of easy cases (… horrible… or … awesome …)
- 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 and other semantic effects
  - *This movie doesn't care about cleverness, wit or any other kind of intelligent humor.*
  - *It's not life-affirming – it's vulgar and mean, but I liked it.*

New Model: Recursive Neural Tensor Nets

- Goal
  - A uniform function that composes two vectors
- More expressive than any other RNN so far
- Idea
  - Allow both additive and mediated multiplicative interactions of vectors

New Model: Recursive Neural Tensor Nets



Neural Tensor Layer

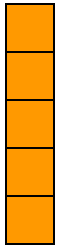
Slices of Tensor Layer

Standard Layer

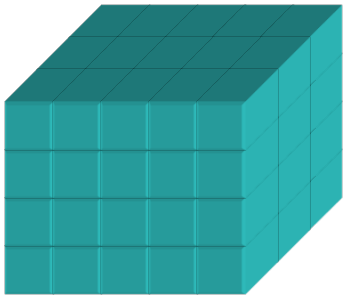
$$p = f\left(\text{[tensor slices]} + \text{[standard layer]} \right)$$

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

Tensor

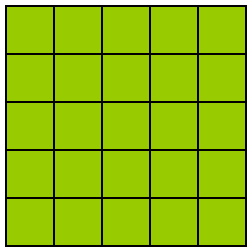


Vector: order-1 tensor



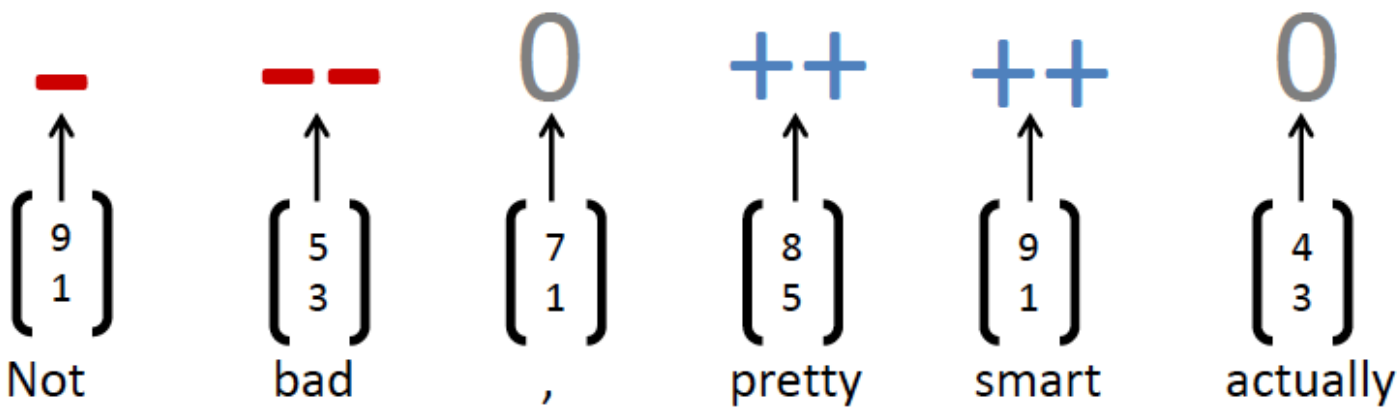
Order-3 tensor

Matrix: order-2 tensor

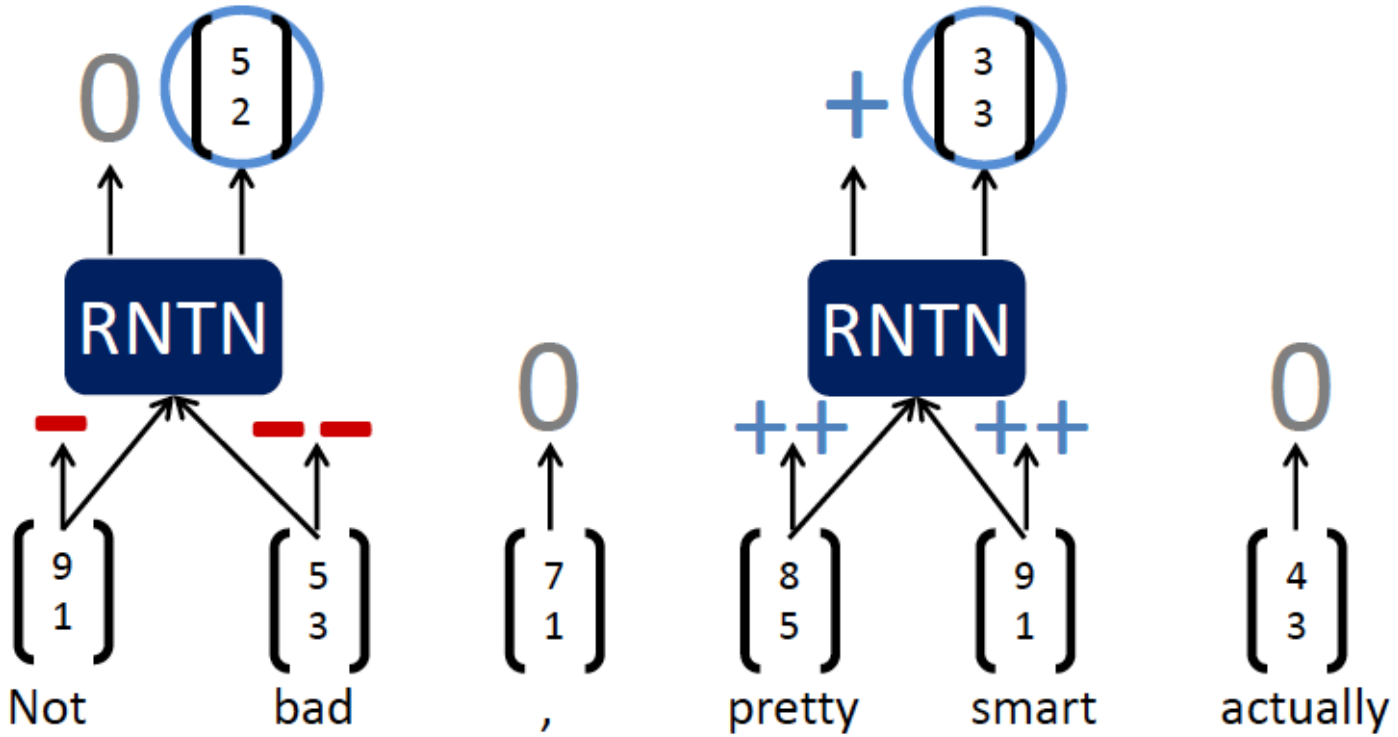


We can interpret each slice of the tensor as capturing a specific type of composition.

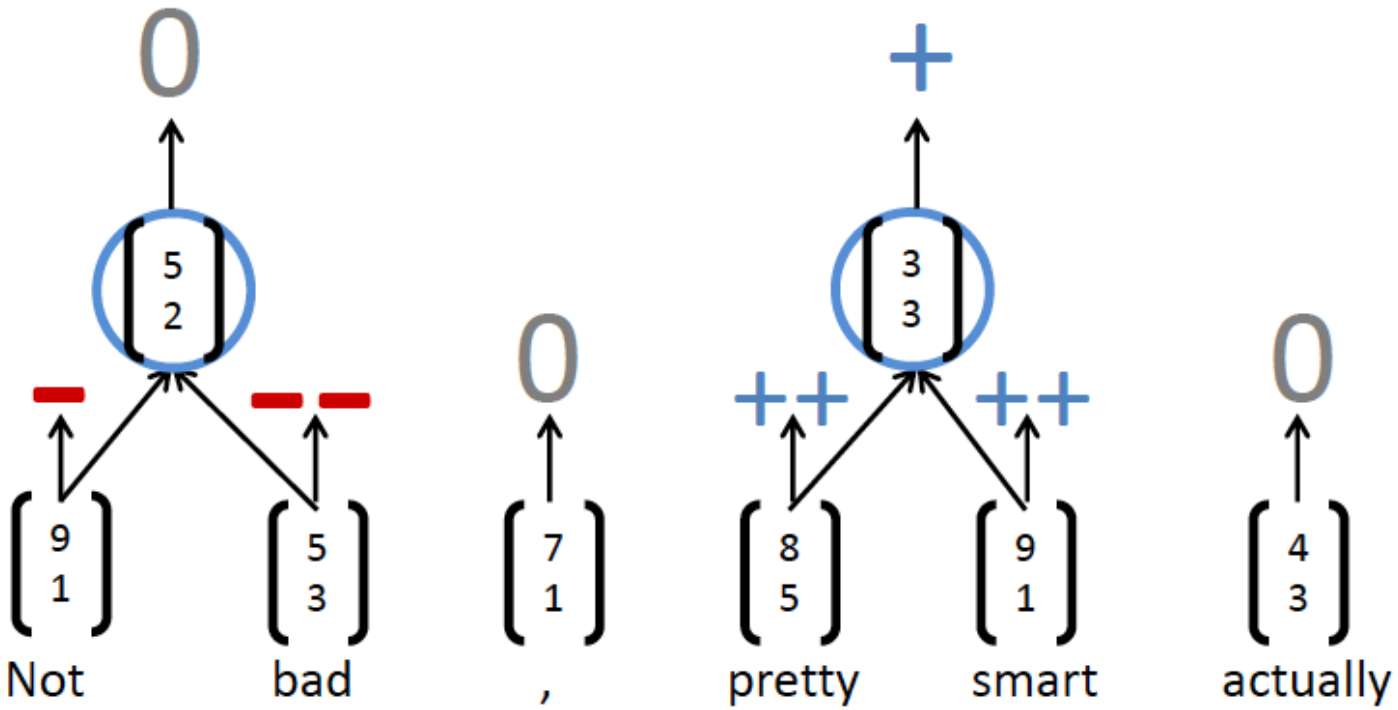
An Example



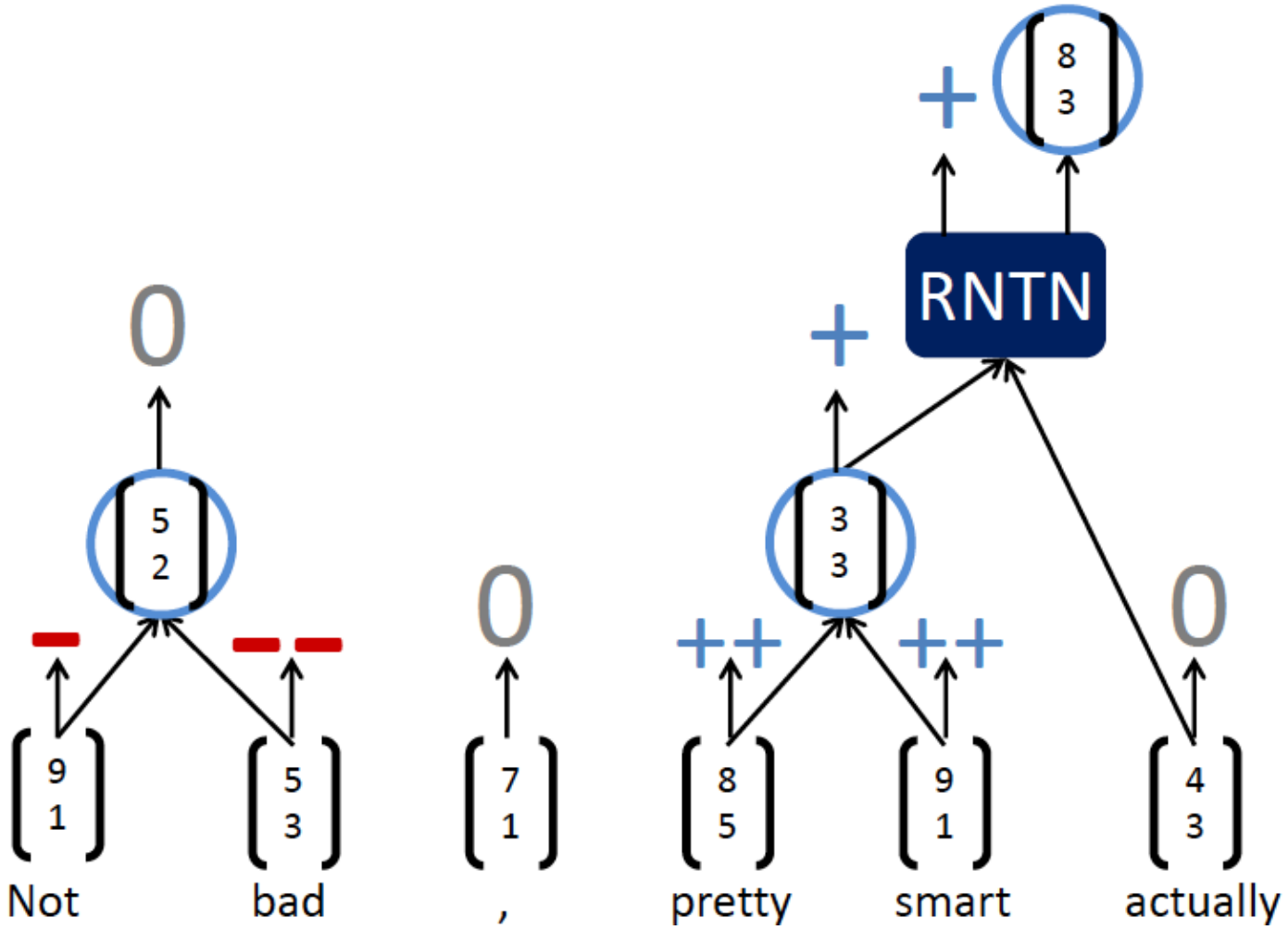
An Example


0
5
2
RNTN
9
1
5
3
Not
bad
0
7
1
,
+
3
3
RNTN
8
5
9
1
pretty
smart
0
4
3
actually

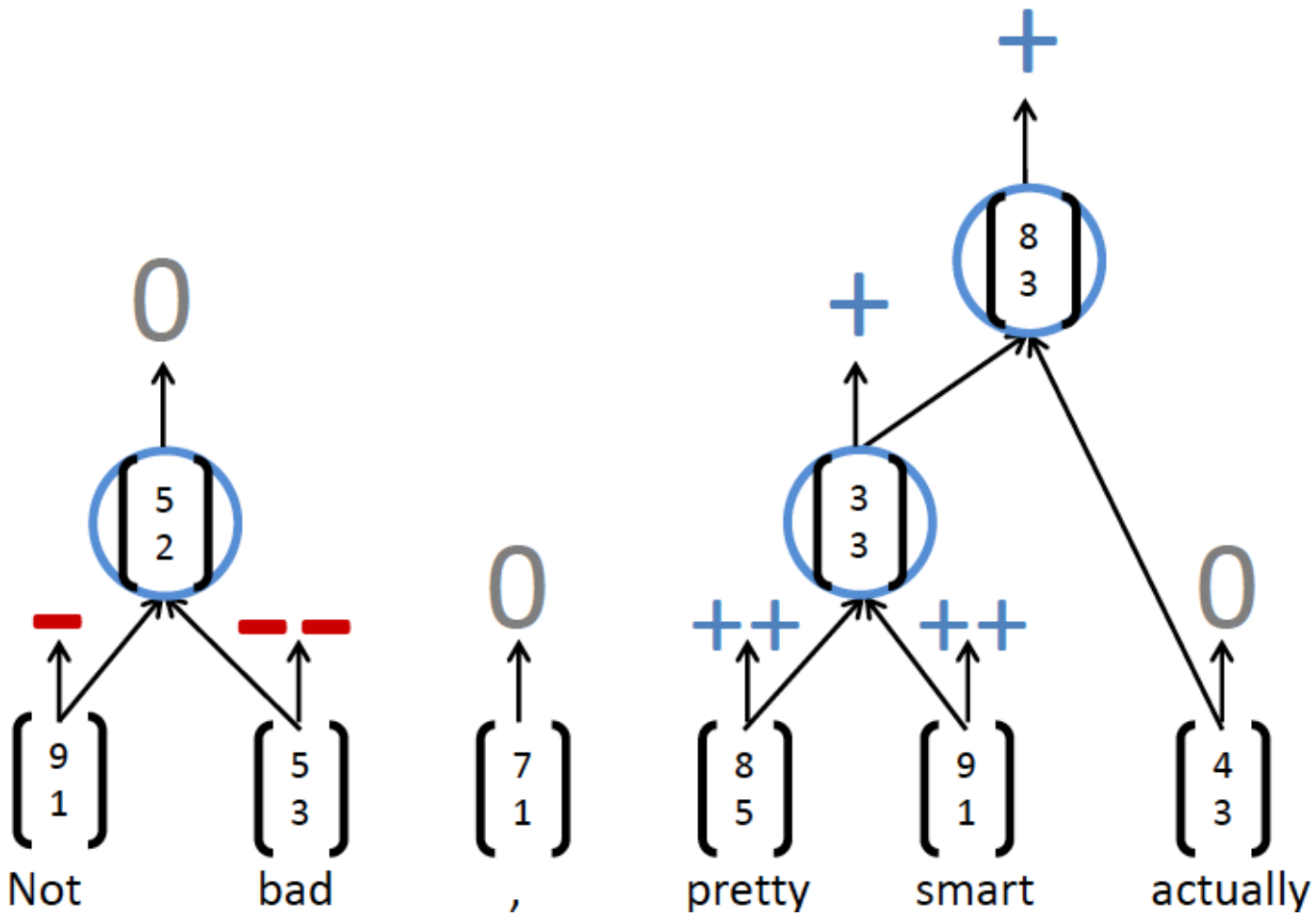
An Example

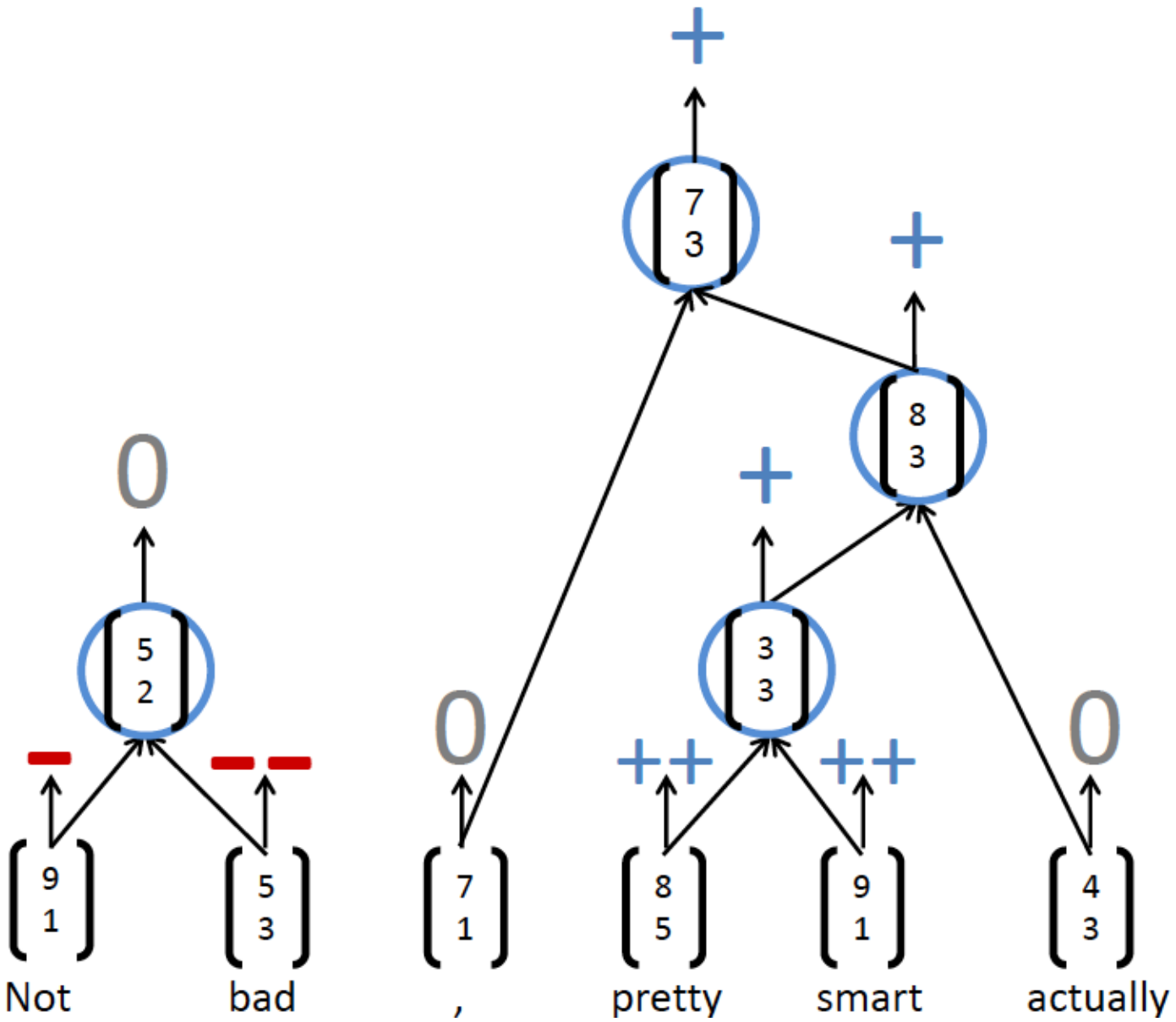


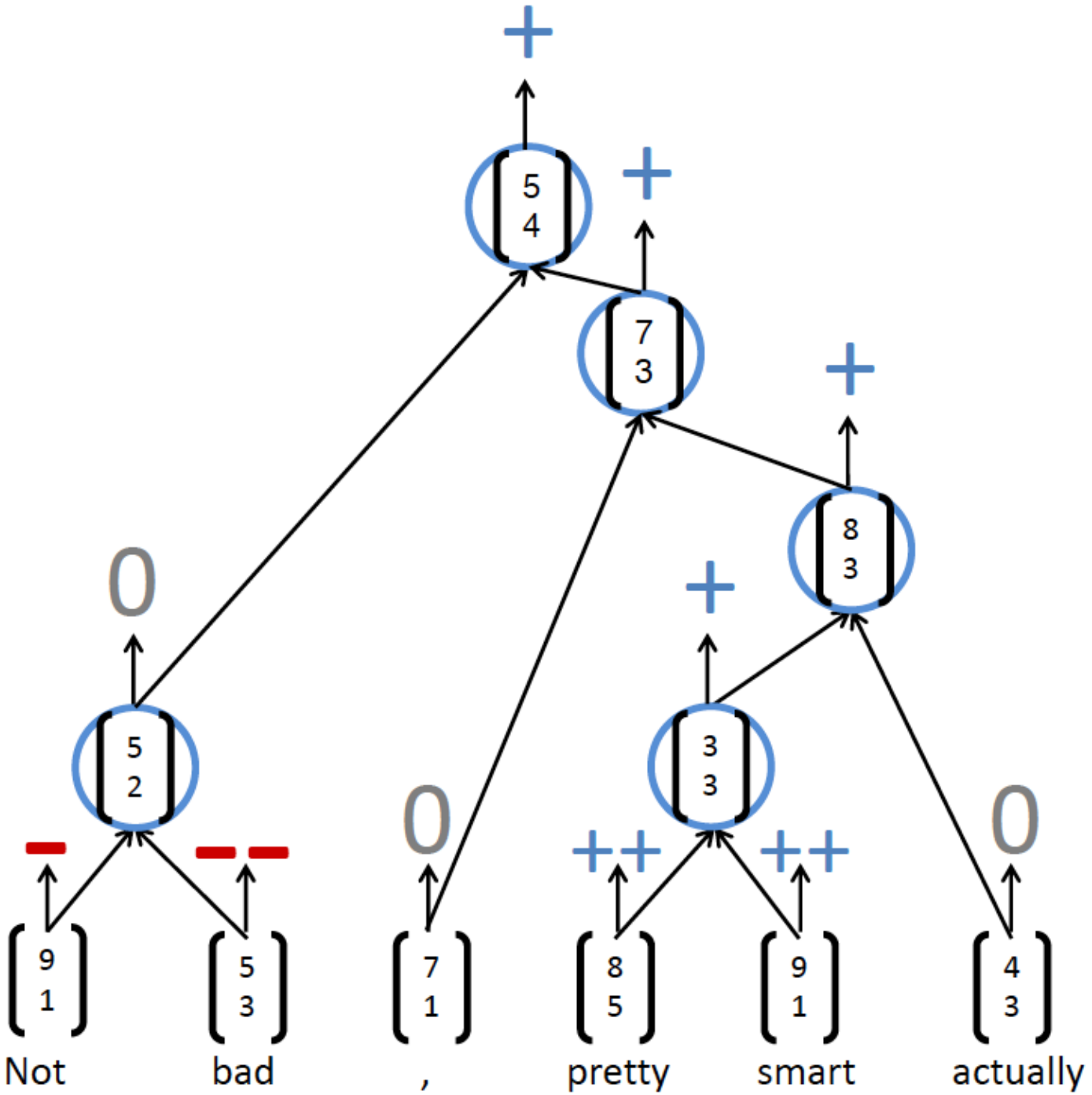
An Example



An Example




0
5
2
9
1
5
3
Not
bad
0
7
3
+
8
3
+
3
3
+
7
1
8
5
9
1
4
3
,
pretty
smart
actually


+
5
4
+
7
3
+
8
3
0
+
5
2
3
3
0
0
++
++
9
1
5
3
7
1
8
5
9
1
4
3
Not
bad
,
pretty
smart
actually

Results

- Accuracy: 85.4