NLP Applications, Recursive Neural Network

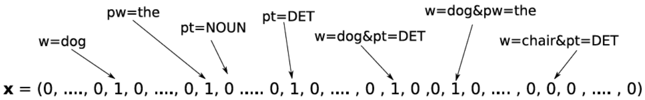
Pawan Goyal

CSE, IIT Kharagpur

March 22-24, 2017

Sequence Labeling Problems

- State-of-the-art models are MaxEnt, CRFs, which use a sparse encoding of features.



Opinion Mining using Deep Recurrent networks: Isroy and Cardie (2014)

Goal

Classify each word as direct subjective expressions (DSEs) and expressive subjective expressions (ESEs)

Opinion Mining using Deep Recurrent networks: Isroy and Cardie (2014)

Goal

Classify each word as direct subjective expressions (DSEs) and expressive subjective expressions (ESEs)

DSE

Explicit mentions of private states or speech events expressing private states

Opinion Mining using Deep Recurrent networks: Isroy and Cardie (2014)

Goal

Classify each word as direct subjective expressions (DSEs) and expressive subjective expressions (ESEs)

DSE

Explicit mentions of private states or speech events expressing private states

ESE

Expressions that indicate sentiment, emotion, etc. without explicitly conveying them

BIO Annotation

BIO Notation

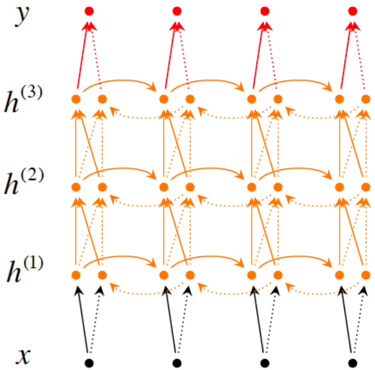
Tags begin-of-entity (B_X), continuation of entity (I_X) or outside (O):

The committee, [as usual]$_{ESE}$, [has refused to make any statements]$_{DSE}$.

The	committee	,	as	usual	,	has
O	O	O	B_ESE	I_ESE	O	B_DSE

refused	to	make	any	statements	.
I_DSE	I_DSE	I_DSE	I_DSE	I_DSE	O

Architecture: Deep Bidirectional RNN



$$\overrightarrow{h}_t^{(i)} = f(\overrightarrow{W}^{(i)} h_t^{(i-1)} + \overrightarrow{V}^{(i)} \overrightarrow{h}_{t-1}^{(i)} + \overrightarrow{b}^{(i)})$$

$$\overleftarrow{h}_t^{(i)} = f(\overleftarrow{W}^{(i)} h_t^{(i-1)} + \overleftarrow{V}^{(i)} \overleftarrow{h}_{t+1}^{(i)} + \overleftarrow{b}^{(i)})$$

$$y_t = g(U[\overrightarrow{h}_t^{(L)}; \overleftarrow{h}_t^{(L)}] + c)$$

Dataset and Performance Metric

- MPQA 1.2 corpus consisting of 535 news articles (11,111 sentences), manually labeled with DSE and ESEs at the phrase level.
- 135 documents for development set for model selection and 10-fold cross validation over the remaining 400 documents.

Dataset and Performance Metric

- MPQA 1.2 corpus consisting of 535 news articles (11,111 sentences), manually labeled with DSE and ESEs at the phrase level.
- 135 documents for development set for model selection and 10-fold cross validation over the remaining 400 documents.

Soft Performance Metric

Dataset and Performance Metric

- MPQA 1.2 corpus consisting of 535 news articles (11,111 sentences), manually labeled with DSE and ESEs at the phrase level.
- 135 documents for development set for model selection and 10-fold cross validation over the remaining 400 documents.

Soft Performance Metric

Binary Overlap: every overlapping match between a predicted and true expression is taken as correct.
Proportional Overlap: imparts a partial correctness, proportional to the overlapping amount, to each match.

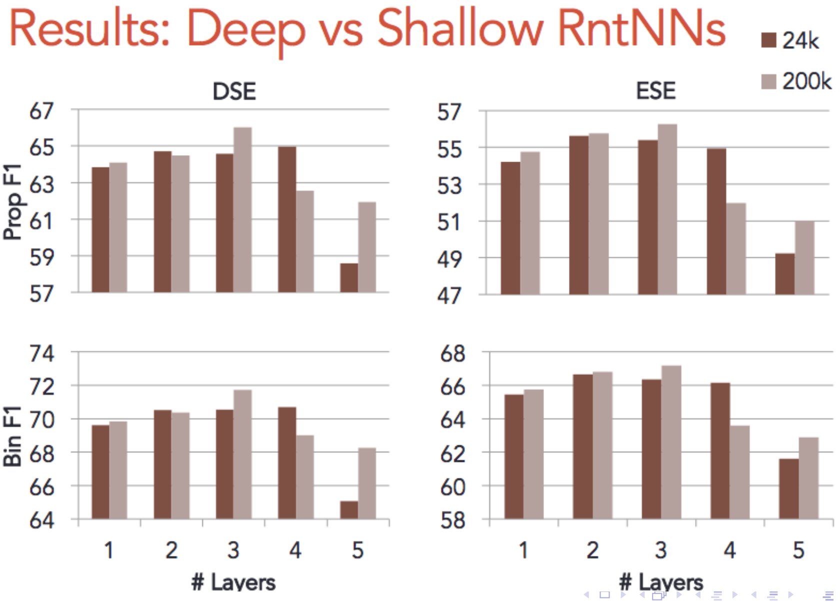
Network Training

- Objective function: standard multiclass cross-entropy
- Learning rate: fixed learning rate (0.005) and a fixed momentum rate (0.7)
- Weight update after minibatches of 80 sentences
- 200 epochs for training
- Select the best model on the development set
- For various models, number of parameters were kept the same.

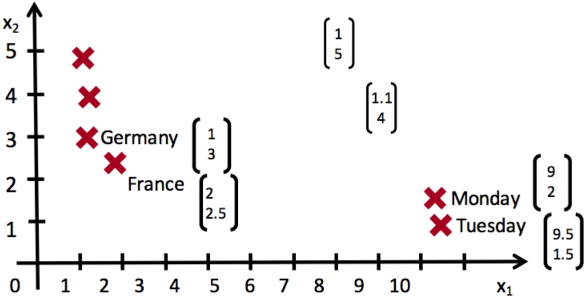
Results: Bidirectional vs. Unidirectional

- Same number of total parameters for each model
- Bidirectional RNN outperforms a unidirectional RNN for both DSE and ESE

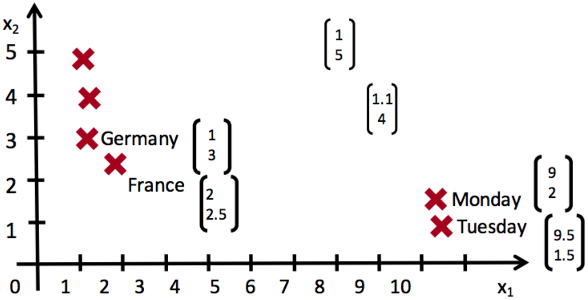
Results: Effect of layers



Next: Building on Word Vector Space Models



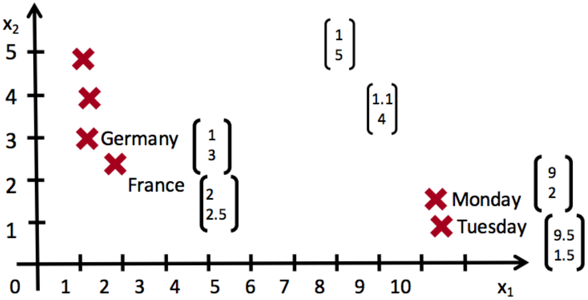
Next: Building on Word Vector Space Models



But how to represent the meaning of long phrases?

- the country of my birth
- the place where I was born

Next: Building on Word Vector Space Models



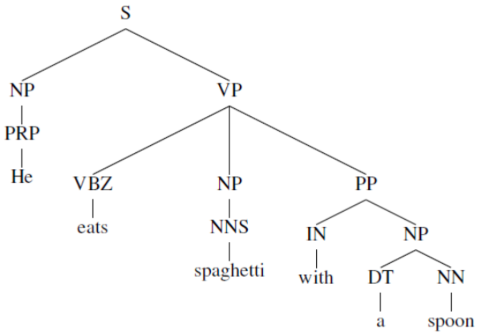
But how to represent the meaning of long phrases?

- the country of my birth
- the place where I was born

Can we map them in the same vector space?

Basic Idea: Recursive Structure in Language

- Recursion helpful in describing language – parse tree.
- Example: "the church which has nice windows", a noun phrase containing a relative clause that contains a noun phrases



Semantic vs. Grammatical Understanding

Semantic Understanding

- Understanding the meaning of the sentence, e.g., being able to represent the phrase as a vector in a structured semantic space
- Similar sentences should be nearby in this space, and unrelated sentences should be far away.

Semantic vs. Grammatical Understanding

Semantic Understanding

- Understanding the meaning of the sentence, e.g., being able to represent the phrase as a vector in a structured semantic space
- Similar sentences should be nearby in this space, and unrelated sentences should be far away.

Grammatical Understanding

- Identify the underlying grammatical structure
- E.g., which part of the sentence depends on which part, what words are modifying what other words, which words act as a single unit etc.
- Usually represented as a parse tree

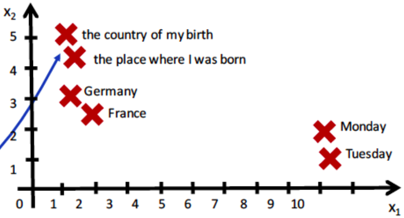
Is grammatical understanding required for semantics?

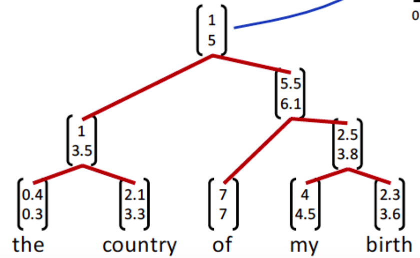
Semantic composition

- First, we need to understand words
- Then, we need to know the way they are put together
- Finally, we can get a meaning of the phrase by leveraging on these two.

Use 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.

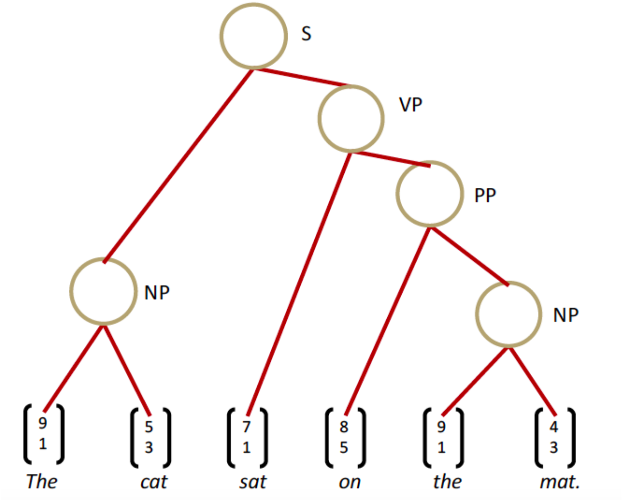




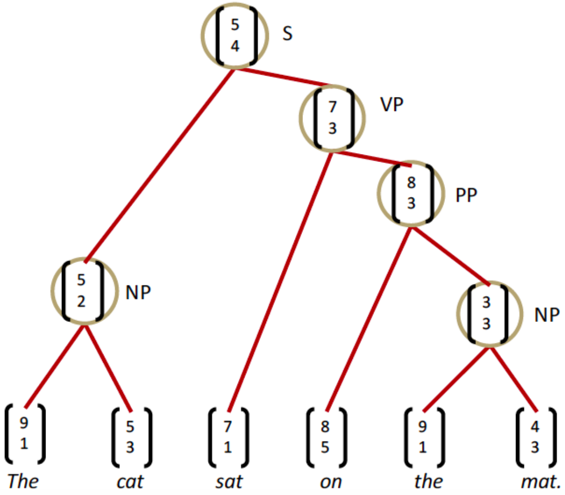
Models in this section can jointly learn parse trees and compositional vector representations

13

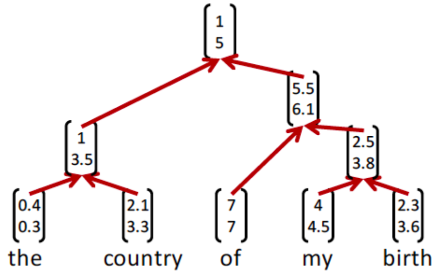
Sentence Parsing: What we want

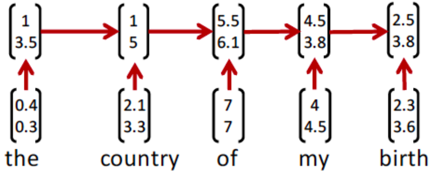


Learn Structure and Representation



Recursive vs. Recurrent Neural Networks





Inputs: Two candidate children's representation

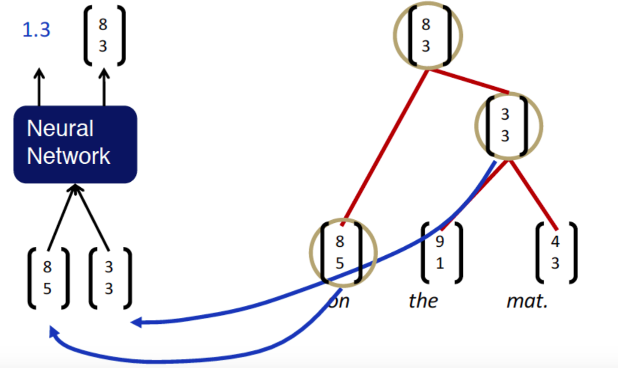
Outputs:

- The semantic representation if the two nodes are merged
- Score of how plausible the new node would be

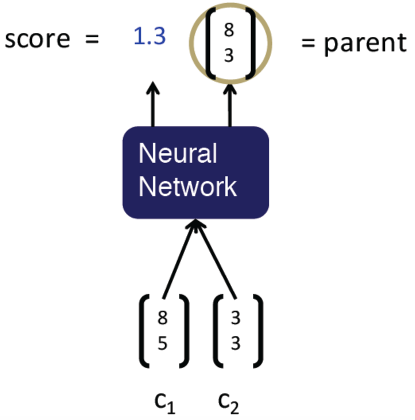
Inputs: Two candidate children's representation

Outputs:

- The semantic representation if the two nodes are merged
- Score of how plausible the new node would be



Recursive Neural Network Definition

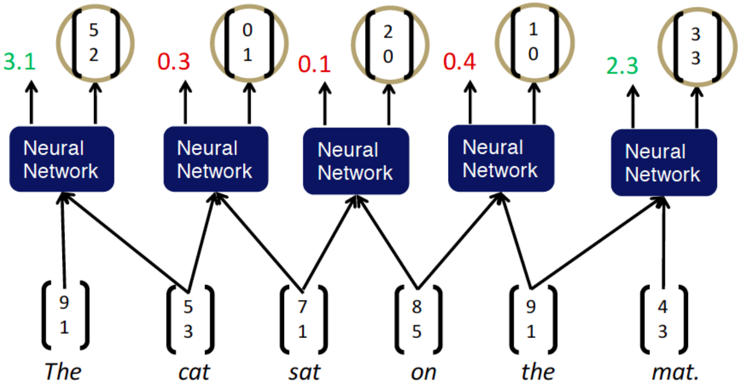


score $= U^{\mathsf{T}} p$

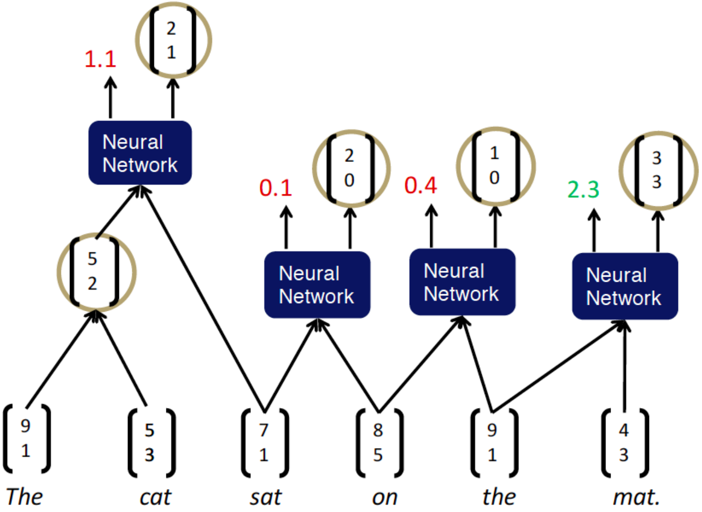
$$\boxed{p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right),}$$

Same W parameters at all nodes of the tree

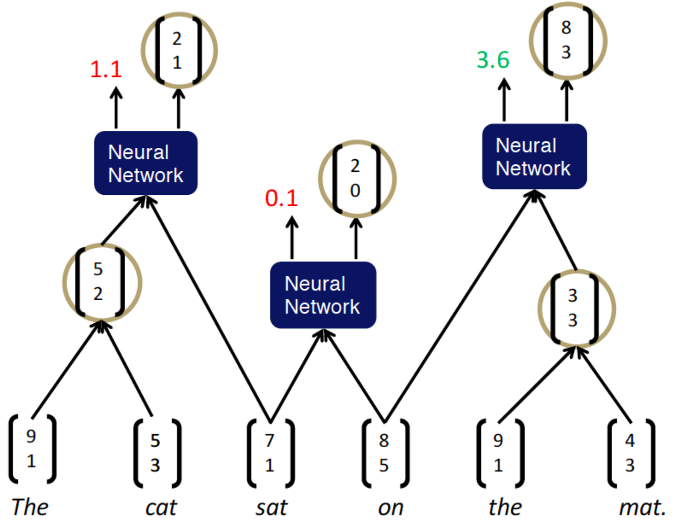
Parsing with Recursive Neural Network



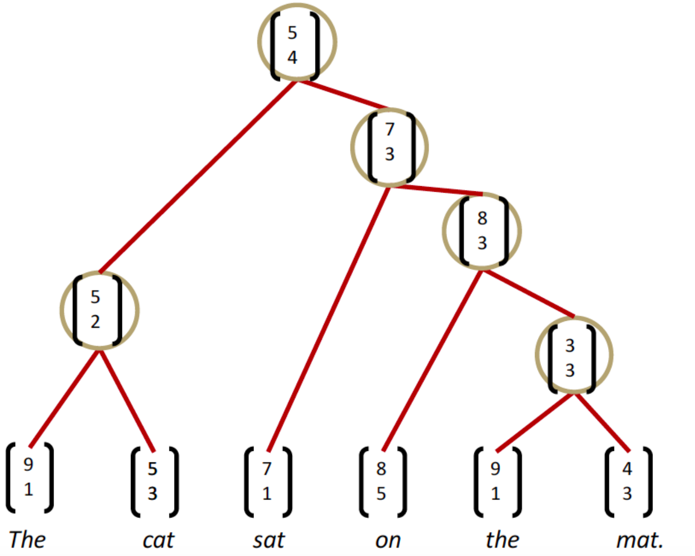
Parsing with Recursive Neural Network



Parsing with Recursive Neural Network



Parsing with Recursive Neural Network

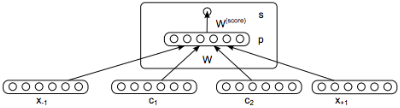


Possible Variations

Context Sensitive RNN

Possible Variations

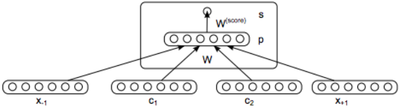
Context Sensitive RNN



$$s = W^{score} p$$
$$p = \tanh(W[x_{-1}; c_1; c_2; x_{+1};] + b^{(1)})$$

Possible Variations

Context Sensitive RNN



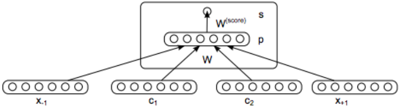
$$
\begin{aligned}
s &= W^{score} p \\
p &= \tanh(W[x_{-1}; c_1; c_2; x_{+1};] + b^{(1)})
\end{aligned}
$$

Category classifier at each non-terminal

- We are only giving a score as output. Can we give category?

Possible Variations

Context Sensitive RNN



$$s = W^{score} p$$
$$p = \tanh(W[x_{-1}; c_1; c_2; x_{+1};] + b^{(1)})$$

Category classifier at each non-terminal

- We are only giving a score as output. Can we give category?
- Remove the scoring layer and use a Softmax layer instead.

Training the Network - Max-Margin Framework

Training the Network - Max-Margin Framework

The score of a tree is computed by the sum of the parsing decisions at each node:

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

Max Margin Objective

The formula used earlier in the course

minimize $J = max(\Delta + s_c - s, 0)$

Max Margin Objective

The formula used earlier in the course

minimize $J = max(\Delta + s_c - s, 0)$ $\Delta = 1$
We would want error to be calculated if $(s - s_c < \Delta)$ and not just when $(s - s_c < 0)$.

Max Margin Objective

The formula used earlier in the course

minimize $J = max(\Delta + s_c - s, 0)$ $\Delta = 1$
We would want error to be calculated if $(s - s_c < \Delta)$ and not just when $(s - s_c < 0)$.

Think of the loss function for the new settings

- We have a gold standard parse tree
- The Recursive network can result in either this or any other parse tree
- *Some parse trees might be more close than others.*

Max-Margin Framework - Details

A supervised objective will be:

Max-Margin Framework - Details

A supervised objective will be:

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

Max-Margin Framework - Details

A supervised objective will be:

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

- y_i is the ground truth parse tree
- The loss $\Delta(y, y_i)$ penalizes all incorrect decisions.
- Structure search for $A(x)$ was maximally greedy – instead, beam search can be used with charting.

Loss: $\Delta(y, y_i)$

- This loss function is a penalization of incorrect spans and adds a penalization λ to each incorrect decision.
- **Span:** a pair of indices which indicate the left and right most leaf nodes under a node in the tree.
- Let $T(y_i)$ denote the set of spans coming from all non-terminal nodes of the tree.

$$\Delta(y, y_i) = \sum_{d \in T(y)} \lambda \mathbb{1}\{d \notin T(y_i)\}$$

Error Back Propagation Through Structure (BPTS)

Error Back Propagation Through Structure (BPTS)

Principally the same as general backpropagation

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

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

Error Back Propagation Through Structure (BPTS)

Principally the same as general backpropagation

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

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

Main differences: similar to BPTT

- Sum derivatives of W from all nodes
- Add error messages from parent + node itself

Error Back Propagation Through Structure (BPTS)

Principally the same as general backpropagation

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

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

Main differences: similar to BPTT

- Sum derivatives of W from all nodes
- Add error messages from parent + node itself

Additional difference

Split derivatives at each node

Some example similarities at phrase level

Center Phrase and Nearest Neighbors
(A) Sales grew almost 2 % to 222.2 million from 222.2 million.
1. Sales surged 22 % to 222.22 billion yen from 222.22 billion.
2. Revenue fell 2 % to 2.22 billion from 2.22 billion.
3. Sales rose more than 2 % to 22.2 million from 22.2 million.
4. Volume was 222.2 million shares, more than triple recent levels.
(B) I had calls all night long from the States , he said.
1. Our intent is to promote the best alternative, he says.
2. We have sufficient cash flow to handle that, he said.
3. Currently, average pay for machinists is 22.22 an hour, Boeing said.
4. Profit from trading for its own account dropped, the securities firm said.

Some example similarities at phrase level

(D) Hess declined to comment.	**(E)** Columbia, S.C
1. PaineWebber declined to comment.	1. Greenville, Miss
2. Phoenix declined to comment.	2. UNK, Md
3. Campeau declined to comment.	3. UNK, Miss
4. Coastal would n't disclose the terms.	4. UNK, Calif
(F) Fujisawa gained 22 to 2,222	**(G)** We were lucky
1. Mochida advanced 22 to 2,222.	1. It was chaotic
2. Commerzbank gained 2 to 222.2.	2. We were wrong
3. Paris loved her at first sight.	3. People had died
4. Profits improved across Hess 's businesses.	4. They still are

Simple Recursive NN: Limitation

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

Solution

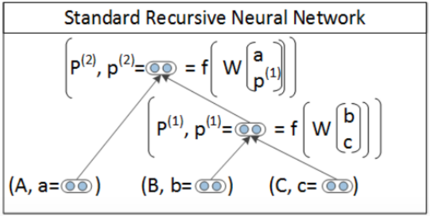
Simple Recursive NN: Limitation

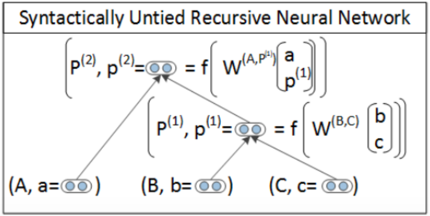
The composition function is the same for all syntactic categories, punctuation, etc.

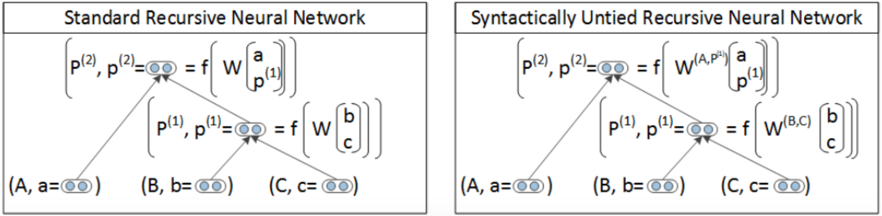
Solution

- Condition the composition function on the syntactic categories
- Allow for different composition functions for pairs of syntactic categories, e.g., Adv + Adj, VP + NP

Syntactically Untied - RNN

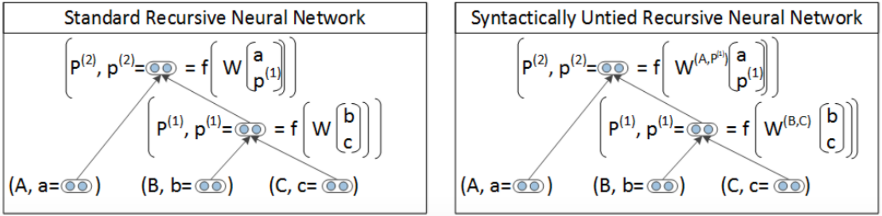






Problem: Every candidate score in bem search needs a matrix-vector product.

Syntactically Untied - RNN



Problem: Every candidate score in bem search needs a matrix-vector product.
Solution: Compute scores only for a subset of trees coming from a simpler, faster model (PCFG).

Compositional Vector Grammars

Scores at each node computed by combination of PCFGs and SU-RNN

Compositional Vector Grammars

Scores at each node computed by combination of PCFGs and SU-RNN

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