CS224d: Deep NLP

Lecture 10:
Advanced Recursive Neural Networks

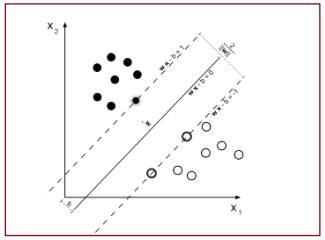
Richard Socher

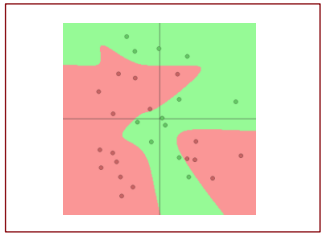
richard@metamind.io

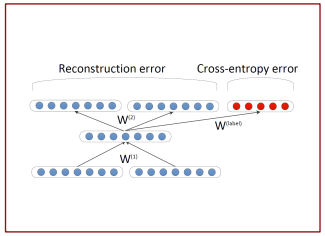
Recursive Neural Networks

- Focused on compositional representation learning of
- Hierarchical structure, features and predictions
- Different combinations of:

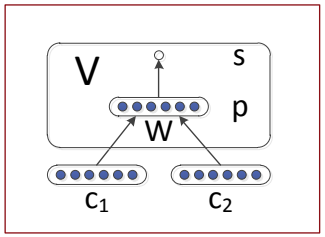
1. Training Objective

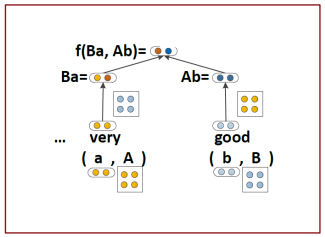


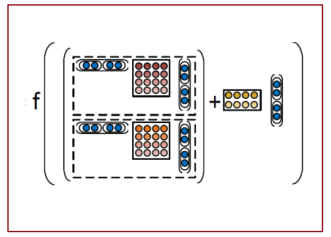




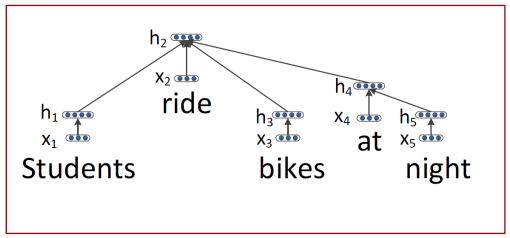
2. **Composition Function**

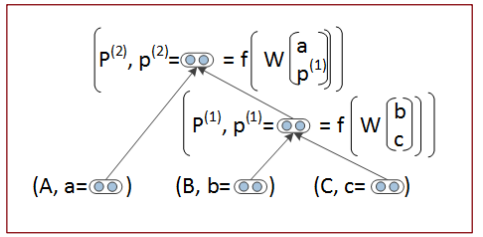






3. Tree Structure





Overview

Last lecture: Recursive Neural Networks

This lecture: Different RNN composition functions and NLP tasks

1. Standard RNNs: Paraphrase detection
2. Matrix-Vector RNNs: Relation classification
3. Recursive Neural Tensor Networks: Sentiment Analysis
4. Tree LSTMs: Phrase Similarity

Next lecture

- Review for Midterm. Going over PSet solutions and common problems/ questions from office hours. Please prepare questions.

Applications and Models

- Note: All models can be applied to all tasks

- More powerful models are needed for harder tasks

- Models get increasingly more expressive and powerful:

1. Standard RNNs: Paraphrase detection
2. Matrix-Vector RNNs: Relation classification
3. Recursive Neural Tensor Networks: Sentiment Analysis
4. Tree LSTMs: Phrase Similarity

Paraphrase Detection

Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses

Basically , the plaintiffs did not show that omissions in Merrill's research caused the claimed losses

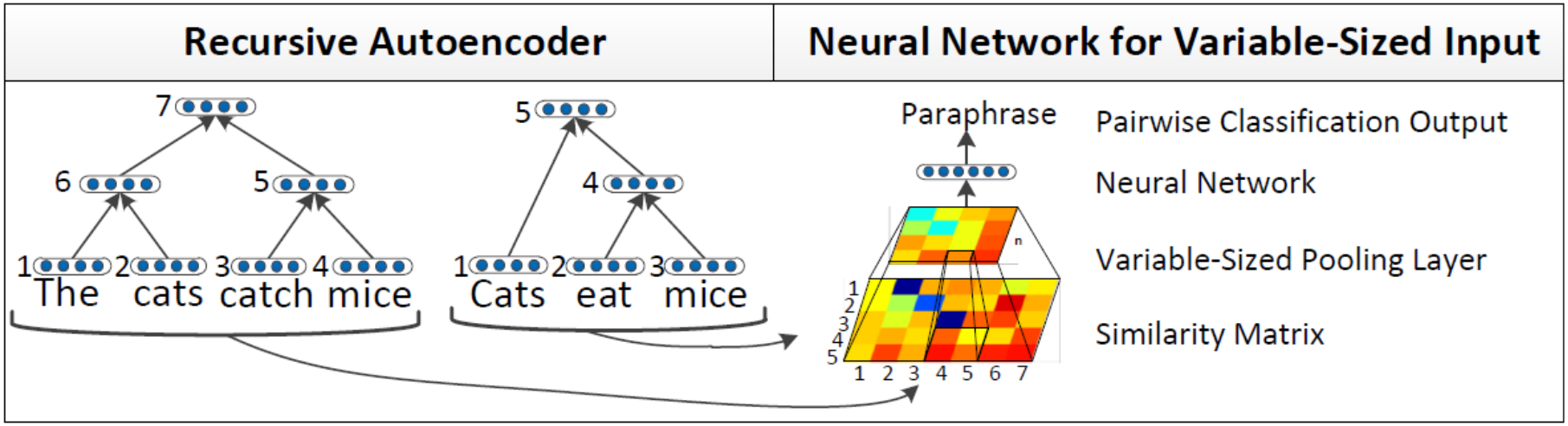
The initial report was made to Modesto Police December 28

It stems from a Modesto police report

How to compare the meaning of two sentences?

RNNs for Paraphrase Detection

Unsupervised RNNs and a pair-wise sentence comparison of nodes in parsed trees (Socher et al., NIPS 2011)



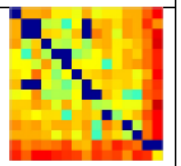
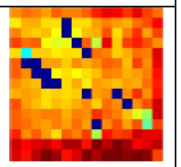
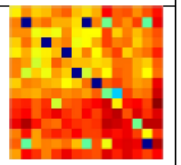
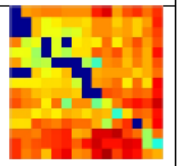
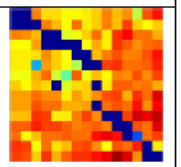
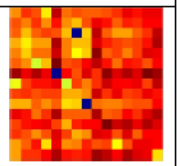
RNNs for Paraphrase Detection

Experiments on Microsoft Research Paraphrase Corpus (Dolan et al. 2004)

Method	Acc.	F1
Rus et al.(2008)	70.6	80.5
Mihalcea et al.(2006)	70.3	81.3
Islam et al.(2007)	72.6	81.3
Qiu et al.(2006)	72.0	81.6
Fernando et al.(2008)	74.1	82.4
Wan et al.(2006)	75.6	83.0
Das and Smith (2009)	73.9	82.3
Das and Smith (2009) + 18 Surface Features	76.1	82.7
F. Bu et al. (ACL 2012): String Re-writing Kernel	76.3	--
Unfolding Recursive Autoencoder (NIPS 2011)	**76.8**	**83.6**

Dataset is problematic, a better evaluation is introduced later

RNNs for Paraphrase Detection

L	Pr	Sentences	Sim.Mat.
P	0.95	(1) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion - Australian football - as the world champion relaxed before his Wimbledon title defence (2) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion- Australian rules football-as the world champion relaxed ahead of his Wimbledon defence	
P	0.82	(1) The lies and deceptions from Saddam have been well documented over 12 years (2) It has been well documented over 12 years of lies and deception from Saddam	
P	0.67	(1) Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses (2) Basically , the plaintiffs did not show that omissions in Merrill's research caused the claimed losses	
N	0.49	(1) Prof Sally Baldwin, 63, from York, fell into a cavity which opened up when the structure collapsed at Tiburtina station, Italian railway officials said (2) Sally Baldwin, from York, was killed instantly when a walkway collapsed and she fell into the machinery at Tiburtina station	
N	0.44	(1) Bremer, 61, is a onetime assistant to former Secretaries of State William P. Rogers and Henry Kissinger and was ambassador-at-large for counterterrorism from 1986 to 1989 (2) Bremer, 61, is a former assistant to former Secretaries of State William P. Rogers and Henry Kissinger	
N	0.11	(1) The initial report was made to Modesto Police December 28 (2) It stems from a Modesto police report	

Recursive Deep Learning

1. Standard RNNs:
2. Matrix-Vector RNNs:
3. Recursive Neural Tensor Networks:
4. Tree LSTMs:

Paraphrase Detection

Relation classification

Sentiment Analysis

Phrase Similarity

Compositionality Through Recursive Matrix-Vector Spaces

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

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

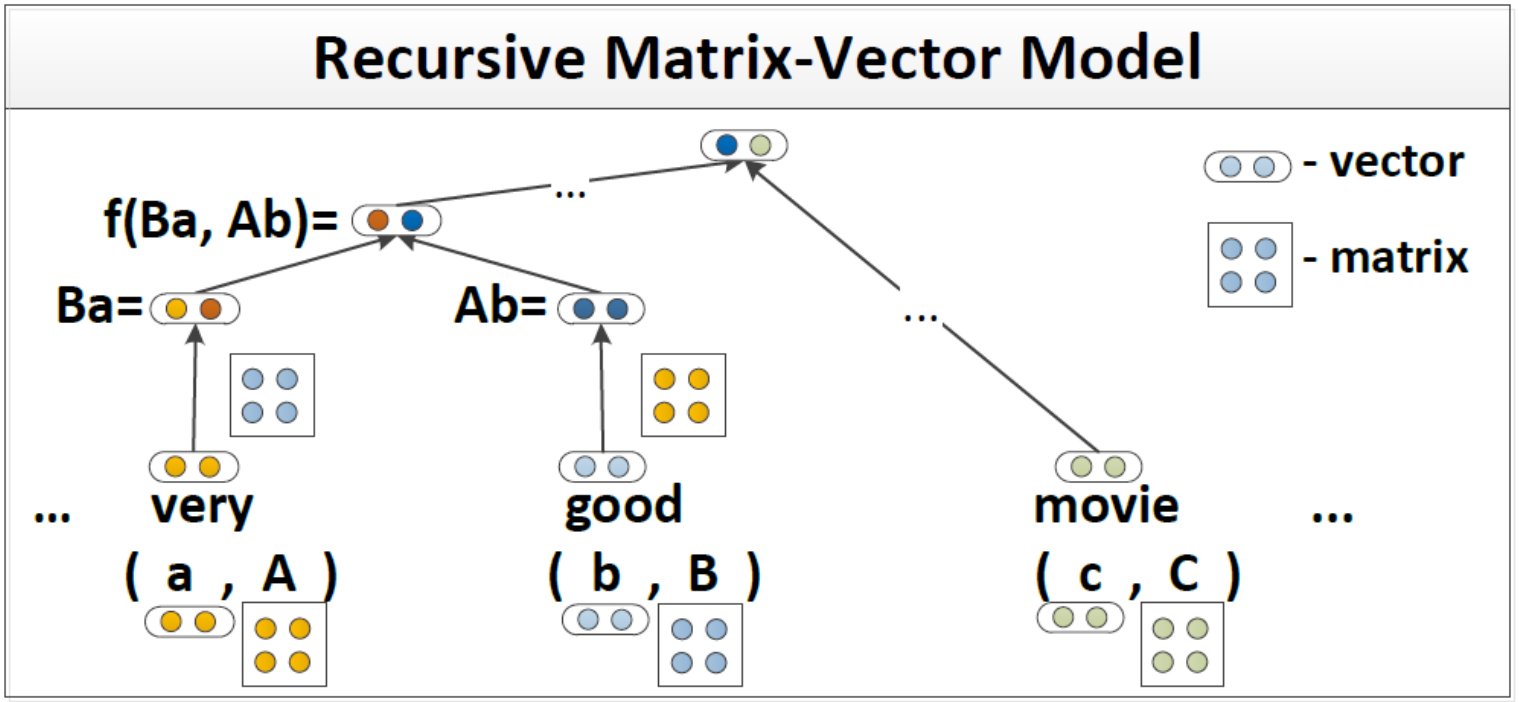
very good

Proposal: A new composition function

Compositionality Through Recursive Matrix-Vector Recursive Neural Networks

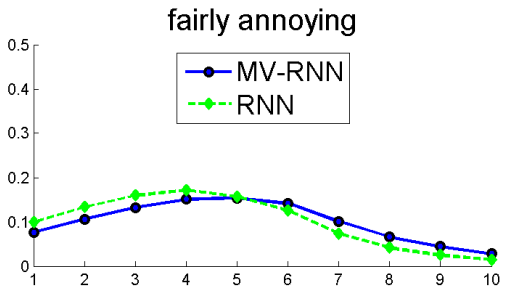
$$p = \tanh\left(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b\right)$$

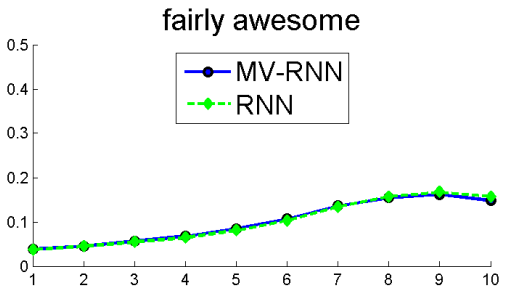
$$p = \tanh\left(W \begin{bmatrix} C_2 c_1 \\ C_1 c_2 \end{bmatrix} + b\right)$$

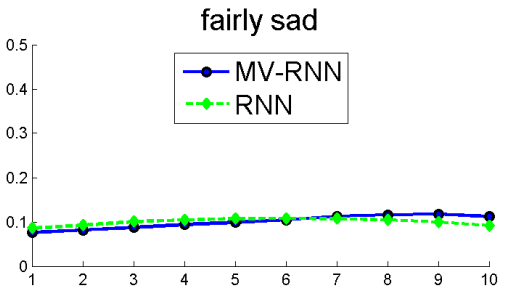


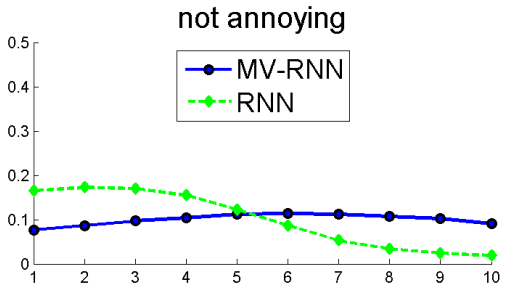
Predicting Sentiment Distributions

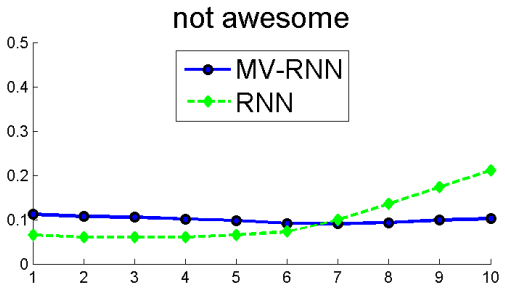
Good example for non-linearity in language

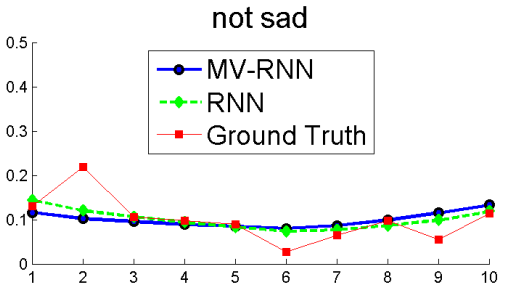


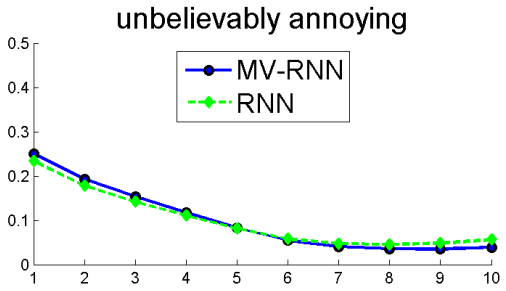


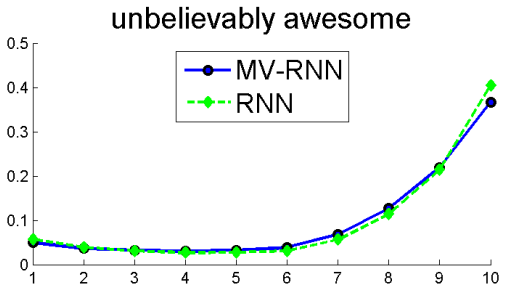


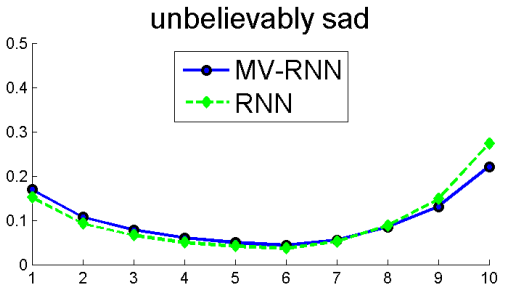




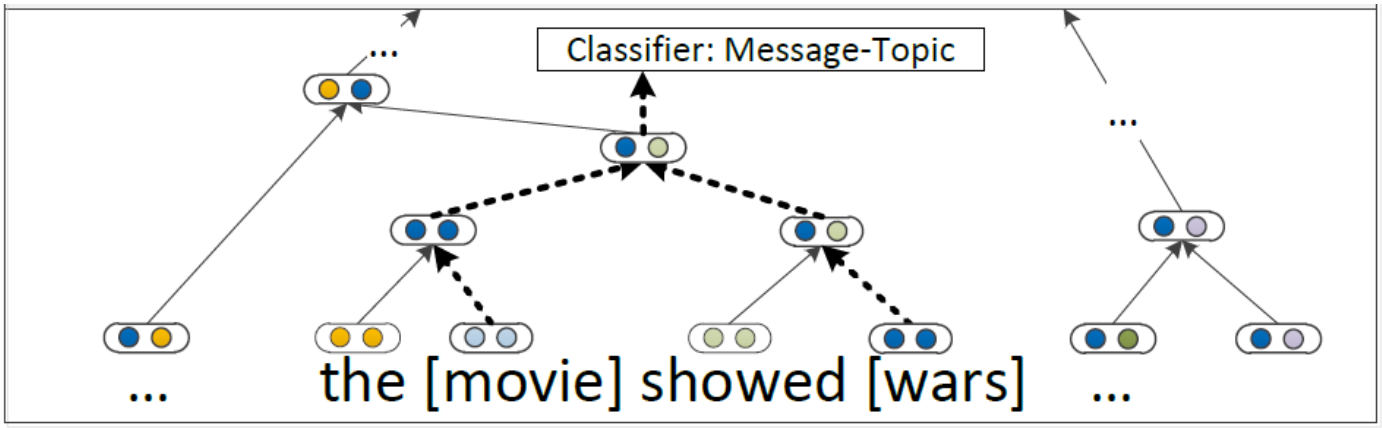








MV-RNN for Relationship Classification



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.

Classifier	Feature Sets	F1
SVM	POS, stemming, syntactic patterns	60.1
SVM	word pair, words in between	72.5
SVM	POS, WordNet, stemming, syntactic patterns	74.8
SVM	POS, WordNet, morphological features, thesauri, Google n-grams	77.6
MaxEnt	POS, WordNet, morphological features, noun compound system, thesauri, Google n-grams	77.6
SVM	POS, WordNet, prefixes and other morphological features, POS, dependency parse features, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner	82.2
RNN	-	74.8
Lin.MVR	-	73.0
MV-RNN	-	79.1
RNN	POS,WordNet,NER	77.6
Lin.MVR	POS,WordNet,NER	78.7
MV-RNN	POS,WordNet,NER	**82.4**

Sentiment Detection

Sentiment detection is crucial to business intelligence, stock trading, …

Maybe she'll change her name to Halliburton. Just to see.

3/18/11 at 4:00 PM | 17 Comments

Mentions of the Name 'Anne Hathaway' May Drive Berkshire Hathaway Stock

By Patrick Huguenin

The Huffington Post recently pointed out that whenever Anne Hathaway is in the news, the stock price for Warren Buffett's Berkshire Hathaway goes up. Really. When *Bride Wars* opened, the stock rose 2.61 percent. (*Rachel Getting Married* only kicked it up 0.44 percent, but, you know, that one was so light on plot compared to *Bride Wars*.)

Sentiment Detection and Bag-of-Words Models

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

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

+ white blood cells destroying an infection

− an infection destroying white blood cells

Sentiment Detection and Bag-of-Words Models

- Sentiment is that sentiment is "easy"
- 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** + other semantic effects

Data: Movie Reviews

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

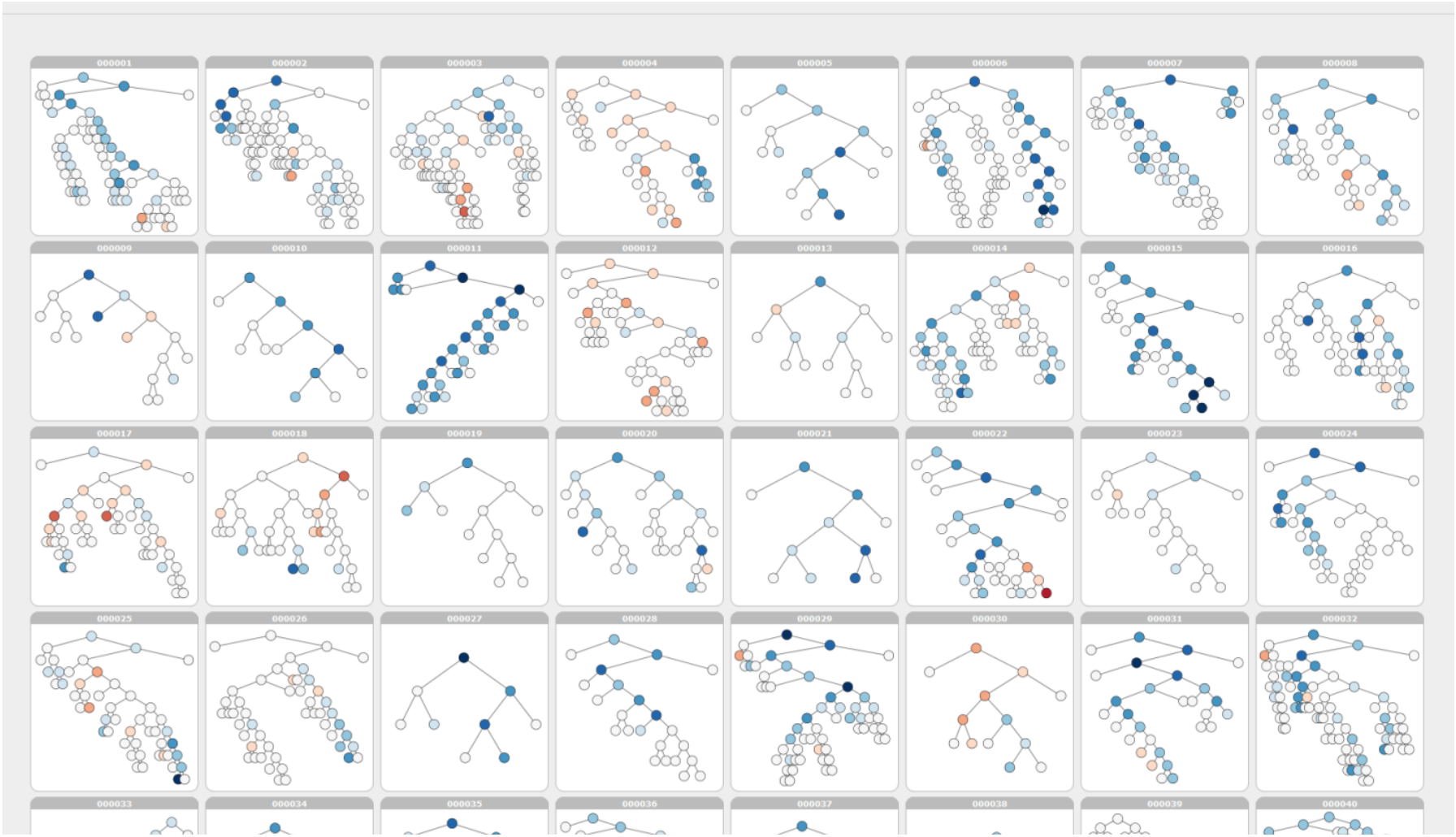
There are slow and repetitive parts but it has just enough spice to keep it interesting.

Two missing pieces for improving sentiment

1. Compositional Training Data

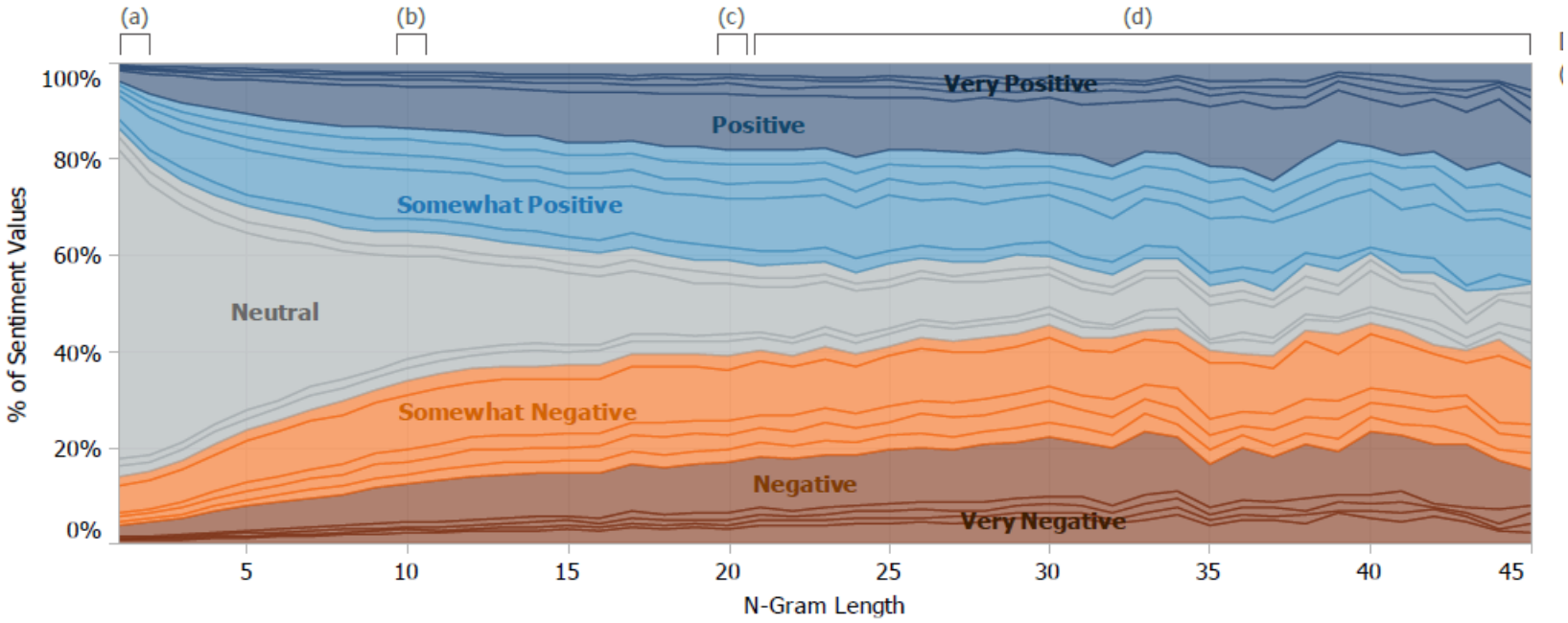
2. Better Compositional model

1. New Sentiment Treebank



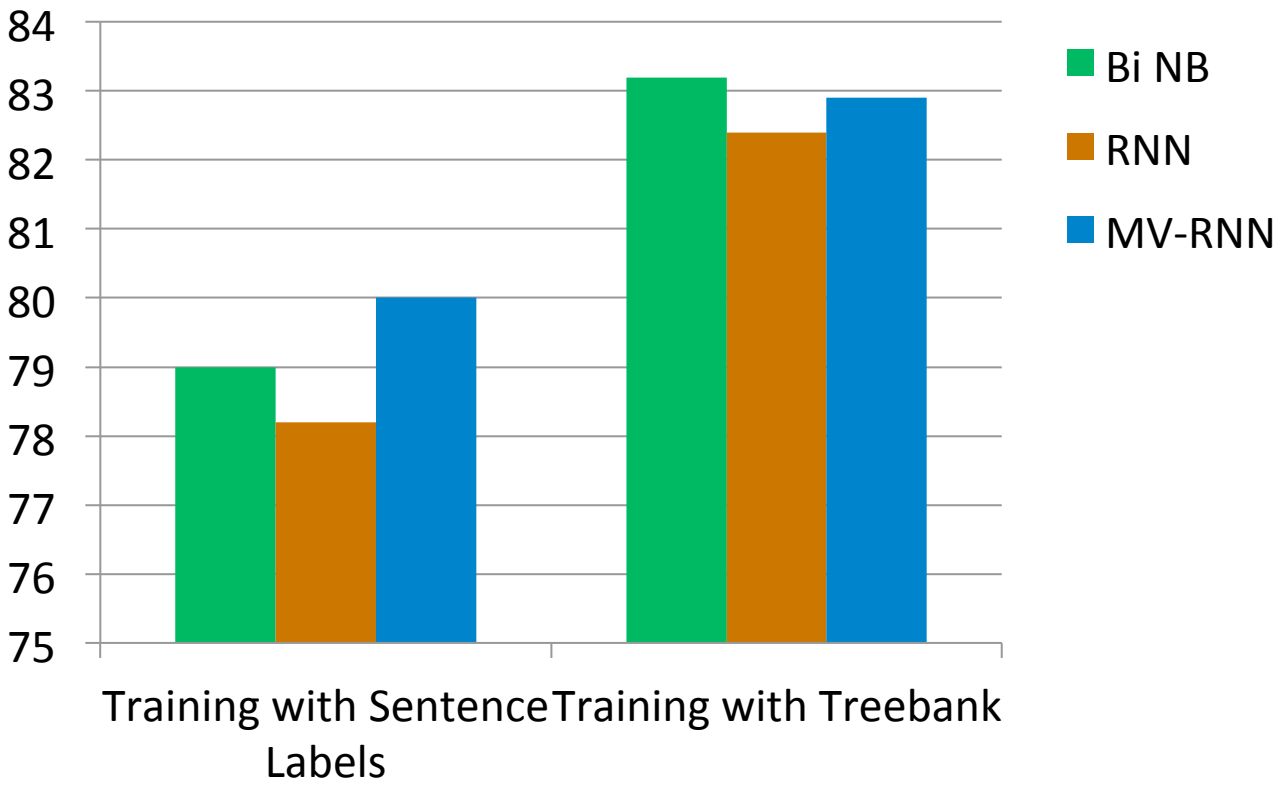
1. New Sentiment Treebank

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



Better Dataset Helped All Models

- Positive/negative full sentence classification



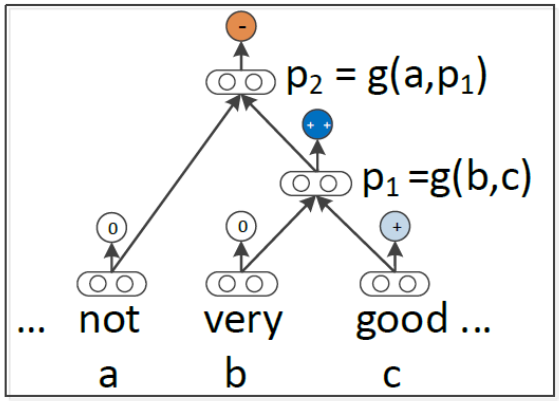
- But hard negation cases are still mostly incorrect
- We also need a more powerful model!

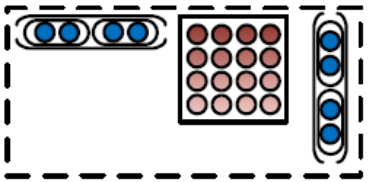
Better Dataset Helped

- This improved performance for full sentence positive/negative classification by 2 – 3 %
- Yay!
- But a more in depth analysis shows: hard negation cases are still mostly incorrect
- We also need a more powerful model!

2. New Compositional Model

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

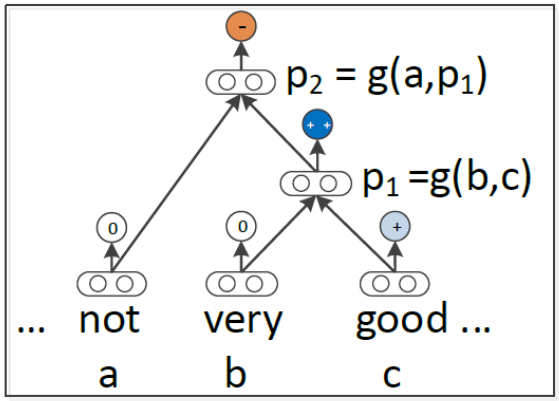


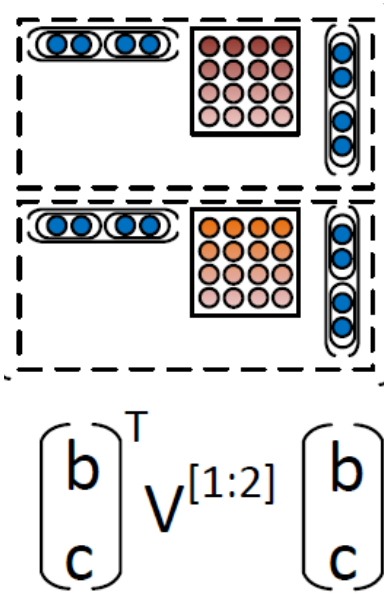


$$\begin{bmatrix} b \\ c \end{bmatrix}^T V \begin{bmatrix} b \\ c \end{bmatrix}$$

2. New Compositional Model

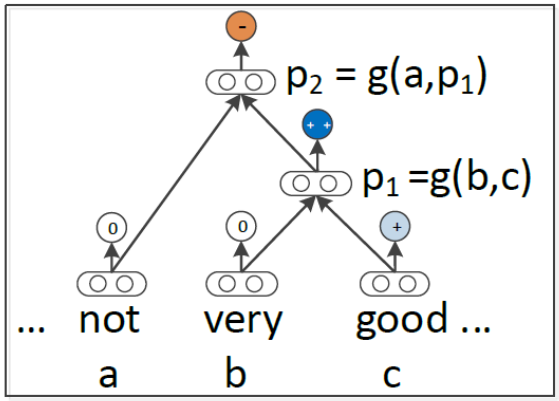
- Recursive Neural Tensor Network





2. New Compositional Model

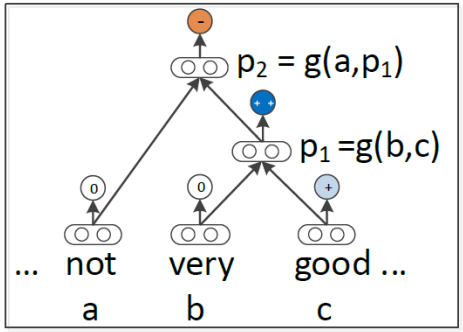
- Recursive Neural Tensor Network



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

Recursive Neural Tensor Network

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



Neural Tensor Layer

Slices of Tensor Layer

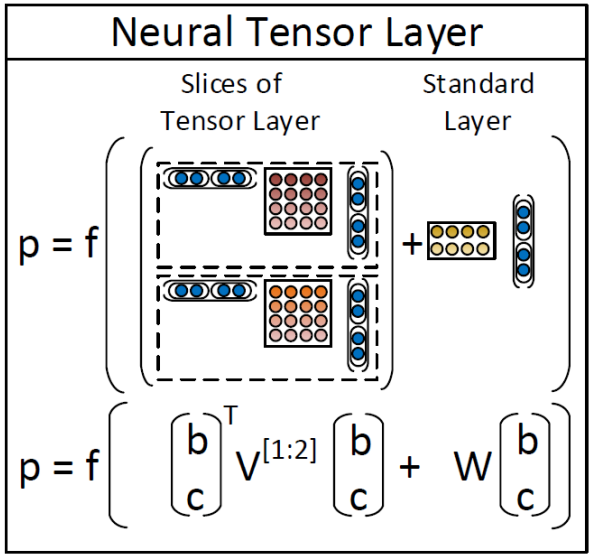
Standard Layer

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

Details: Tensor Backpropagation Training

- Main new matrix derivative needed for a tensor:

$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{a}}{\partial \mathbf{X}} = \frac{\partial \mathbf{a}^T \mathbf{X}^T \mathbf{a}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{a}^T$$

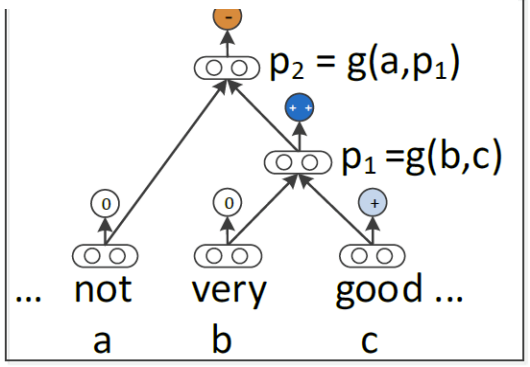


Details: Tensor Backpropagation Training

- Minimizing cross entropy error:

$$E(\theta) = \sum_i \sum_j t_j^i \log y_j^i + \lambda \|\theta\|^2$$

$$y^a = \text{softmax}(W_s a)$$



- Standard softmax error message:

$$\delta^{i,s} = \left(W_s^T (y^i - t^i)\right) \otimes f'(x^i)$$

- For each slice, we have update:

$$\frac{\partial E^{p_2}}{\partial V^{[k]}} = \delta_k^{p_2,com} \begin{bmatrix} a \\ p_1 \end{bmatrix} \begin{bmatrix} a \\ p_1 \end{bmatrix}^T$$

- Main backprop rule to pass error down from parent:

$$\delta^{p_2,down} = \left(W^T \delta^{p_2,com} + S\right) \otimes f'\left(\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

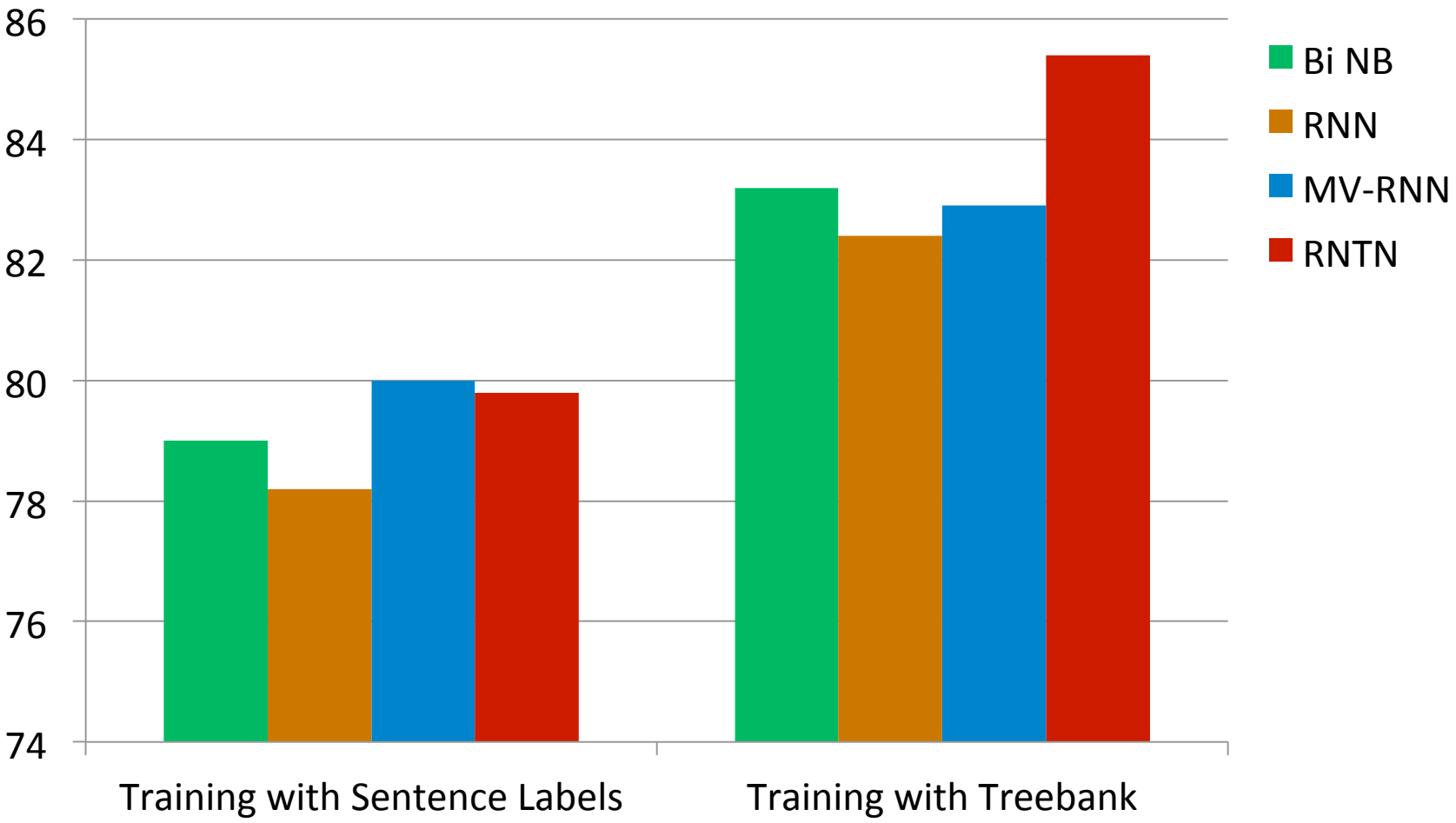
$$S = \sum_{k=1}^{d} \delta_k^{p_2,com} \left(V^{[k]} + \left(V^{[k]}\right)^T\right) \begin{bmatrix} a \\ p_1 \end{bmatrix}$$

- Finally, add errors from parent and current softmax:

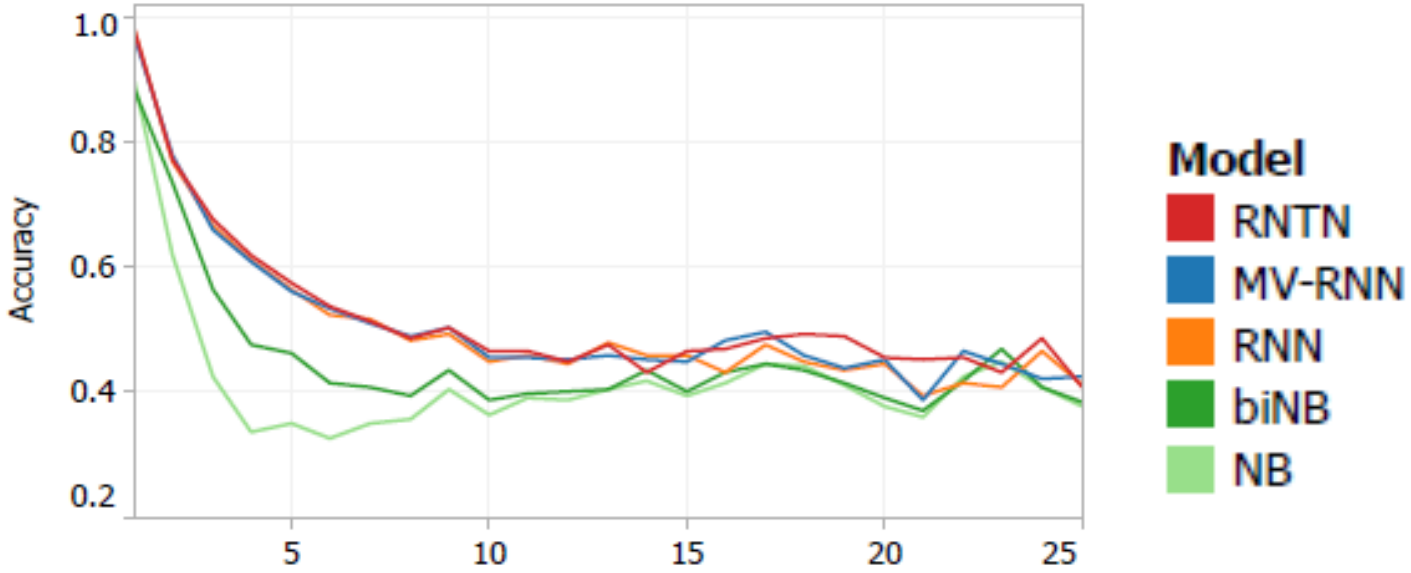
$$\delta^{p_1,com} = \delta^{p_1,s} + \delta^{p_2,down}[d+1:2d]$$

Positive/Negative Results on Treebank

Classifying Sentences: Accuracy improves to 85.4

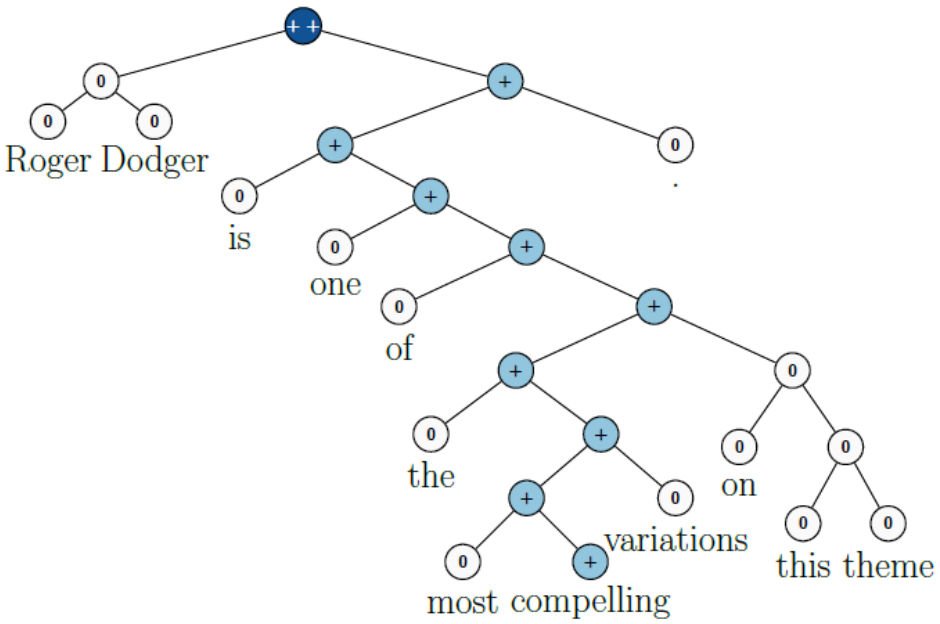


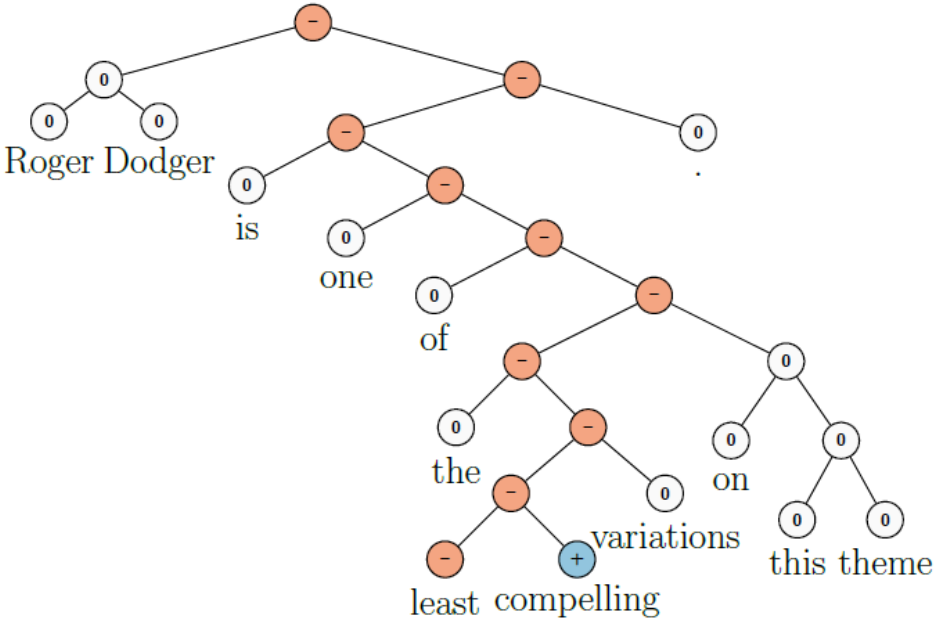
Fine Grained Results on Treebank



Model	Fine-grained	
	All	Root
NB	67.2	41.0
SVM	64.3	40.7
BiNB	71.0	41.9
VecAvg	73.3	32.7
RNN	79.0	43.2
MV-RNN	78.7	44.4
RNTN	**80.7**	**45.6**

Negation Results

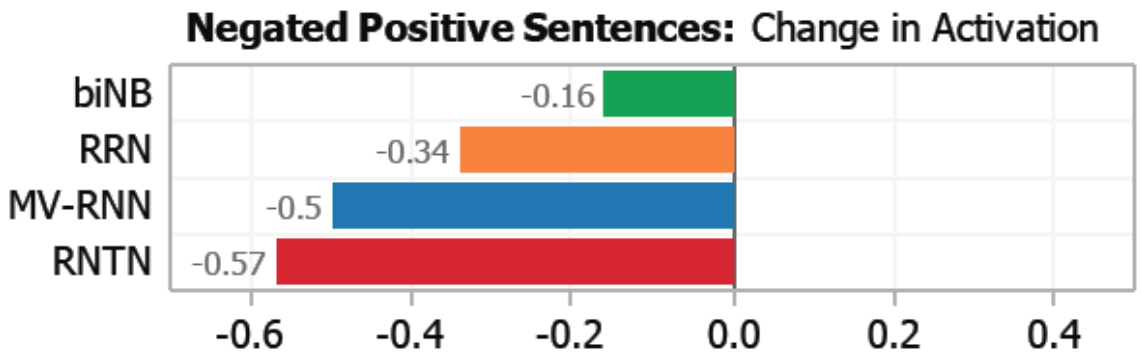




Negation Results

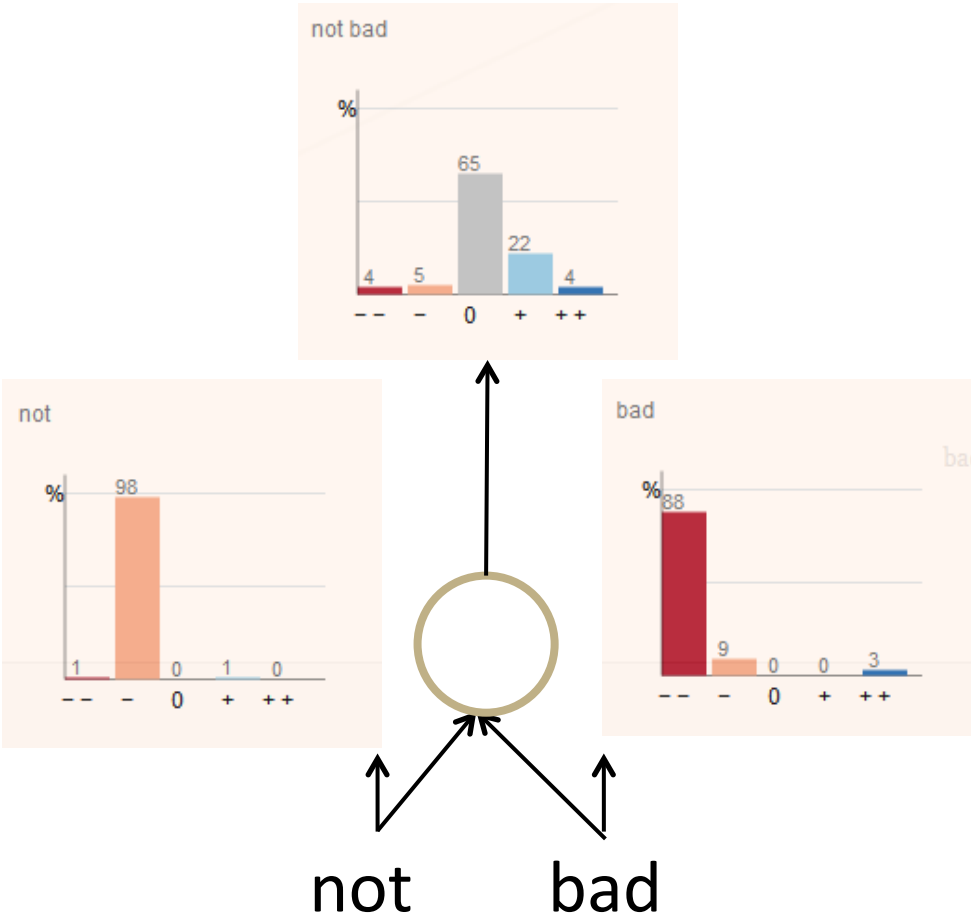
- Most methods capture that negation often makes things more negative (See Potts, 2010)
- Analysis on negation dataset
- Accuracy:

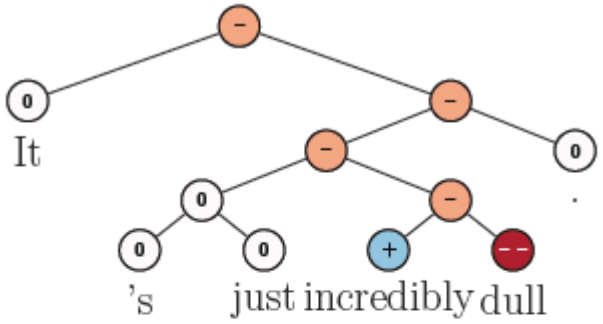
	Negated Positive
biNB	19.0
RNN	33.3
MV-RNN	52.4
RNTN	**71.4**

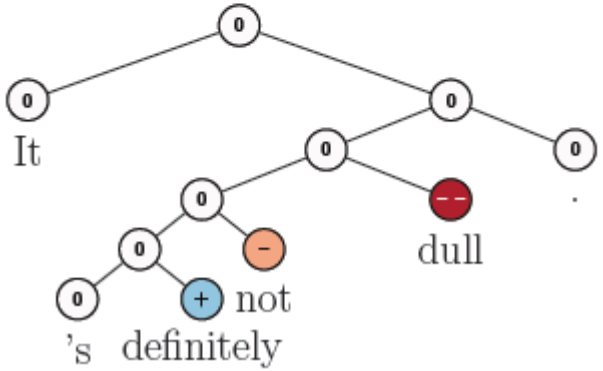


Results on Negating Negatives

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



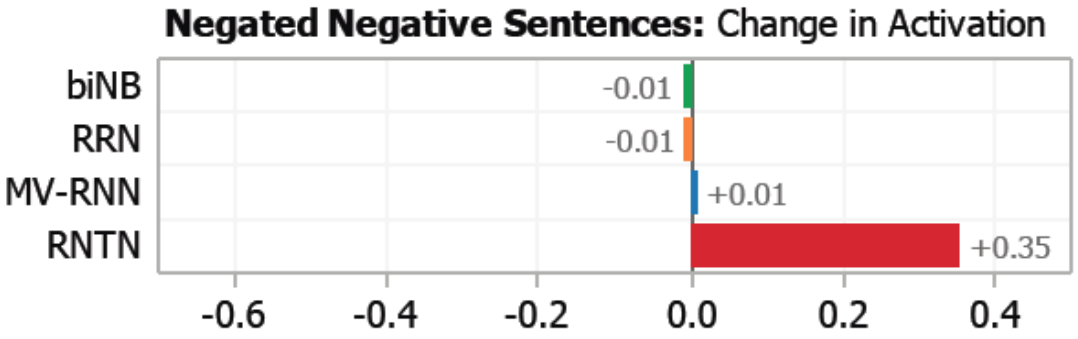




Results on Negating Negatives

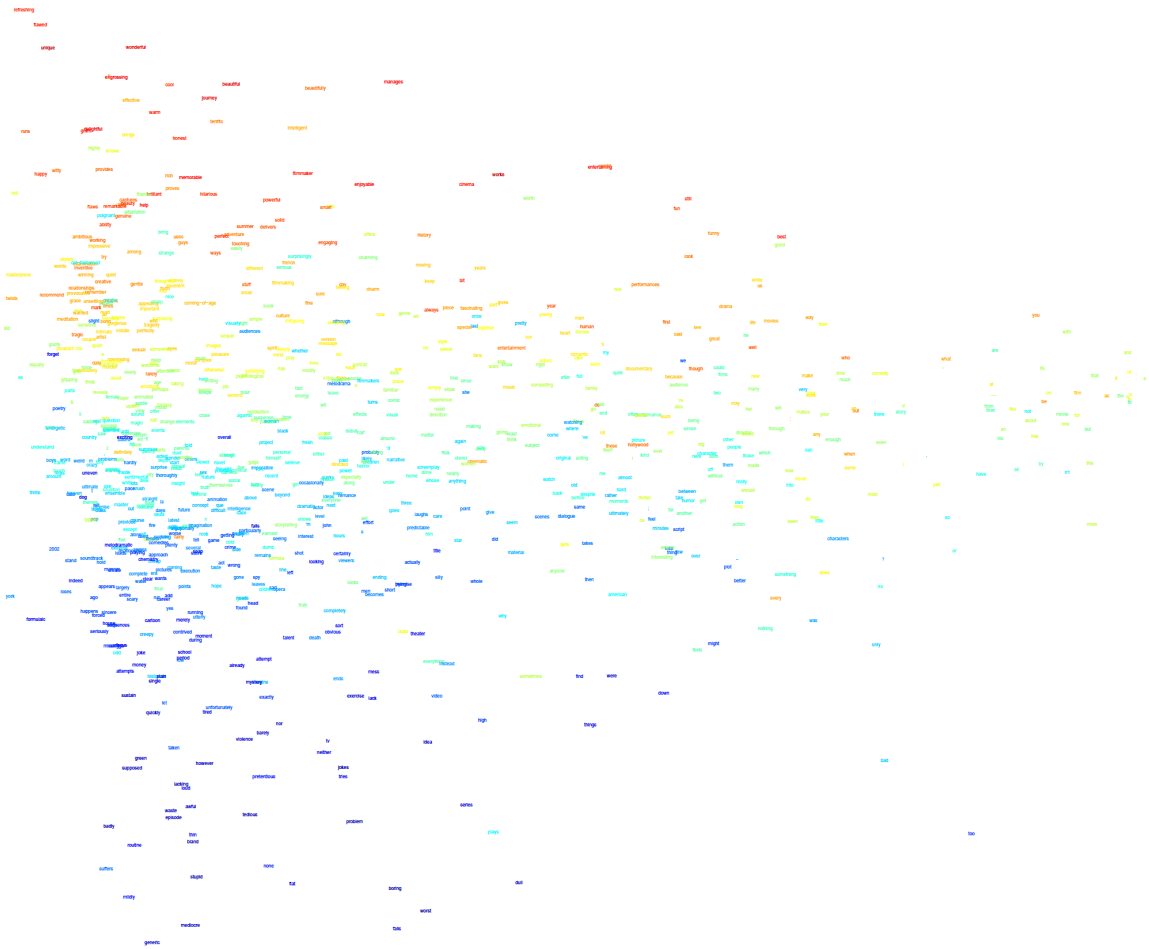
- Evaluation: Positive activation should increase

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**



n	**Most positive n-grams**	**Most negative n-grams**
1	engaging ; best ; powerful ; love ; beautiful ; entertaining ; clever ; terrific ; excellent ; great ;	bad ; dull ; boring ; fails ; worst ; stupid ; painfully ; cheap ; forgettable ; disaster ;
2	excellent performances ; amazing performance ; terrific performances ; A masterpiece ; masterful film ; wonderful film ; terrific performance ; masterful piece ; wonderful movie ; marvelous performances ;	worst movie ; bad movie ; very bad ; shapeless mess ; worst thing ; tepid waste ; instantly forgettable ; bad film ; extremely bad ; complete failure ;
3	an amazing performance ; a terrific performance ; a wonderful film ; wonderful all-ages triumph ; A masterful film ; a wonderful movie ; a tremendous performance ; drawn excellent performances ; most visually stunning ; A stunning piece ;	for worst movie ; A lousy movie ; most joyless movie ; a complete failure ; another bad movie ; fairly terrible movie ; a bad movie ; extremely unfunny film ; most painfully marginal ; very bad sign ;
5	nicely acted and beautifully shot ; gorgeous imagery , effective performances ; the best of the year ; a terrific American sports movie ; very solid , very watchable ; a fine documentary does best ; refreshingly honest and ultimately touching ;	silliest and most incoherent movie ; completely crass and forgettable movie ; just another bad movie . ; drowns out the lousy dialogue ; a fairly terrible movie ... ; A cumbersome and cliche-ridden movie ; a humorless , disjointed mess ;
8	one of the best films of the year ; simply the best family film of the year ; the best film of the year so far ; A love for films shines through each frame ; created a masterful piece of artistry right here ; A masterful film from a master filmmaker , ; 's easily his finest American film ... comes ;	A trashy , exploitative , thoroughly unpleasant experience ; this sloppy drama is an empty vessel . ; a meandering , inarticulate and ultimately disappointing film ; an unimaginative , nasty , glibly cynical piece ; bad , he 's really bad , and ; quickly drags on becoming boring and predictable . ; be the worst special-effects creation of the year ;

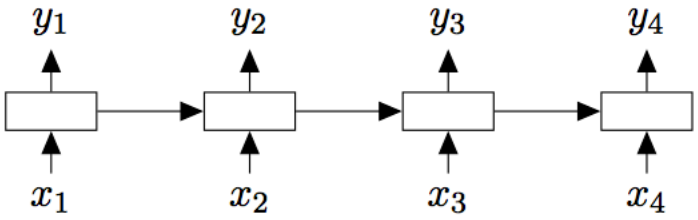
Visualizing Deep Learning: Word Embeddings



LSTMs

- Remember LSTMs?

- Historically only over temporal sequences

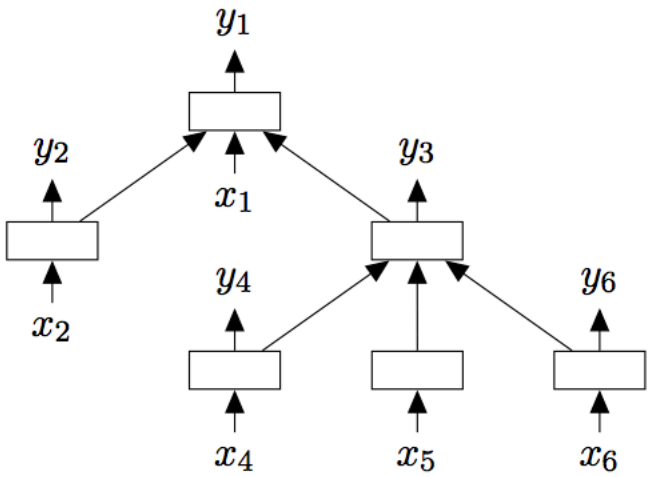


We used $\tilde{c}_t$

$$
\begin{aligned}
i_t &= \sigma\left(W^{(i)}x_t + U^{(i)}h_{t-1} + b^{(i)}\right), \\
f_t &= \sigma\left(W^{(f)}x_t + U^{(f)}h_{t-1} + b^{(f)}\right), \\
o_t &= \sigma\left(W^{(o)}x_t + U^{(o)}h_{t-1} + b^{(o)}\right), \\
u_t &= \tanh\left(W^{(u)}x_t + U^{(u)}h_{t-1} + b^{(u)}\right) \\
c_t &= i_t \odot u_t + f_t \odot c_{t-1}, \\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}
$$

Tree LSTMs

- We can use those ideas in grammatical tree structures!
- Paper: Tai et al. 2015:
  Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks



- Idea: Sum the child vectors in a tree structure
- Each child has its own forget gate
- Same softmax on h

$$\tilde{h}_j = \sum_{k \in C(j)} h_k,$$

$$i_j = \sigma\left(W^{(i)} x_j + U^{(i)} \tilde{h}_j + b^{(i)}\right),$$

$$f_{jk} = \sigma\left(W^{(f)} x_j + U^{(f)} h_k + b^{(f)}\right),$$

$$o_j = \sigma\left(W^{(o)} x_j + U^{(o)} \tilde{h}_j + b^{(o)}\right),$$

$$u_j = \tanh\left(W^{(u)} x_j + U^{(u)} \tilde{h}_j + b^{(u)}\right),$$

$$c_j = i_j \odot u_j + \sum_{k \in C(j)} f_{jk} \odot c_k,$$

$$h_j = o_j \odot \tanh(c_j),$$

Results on Stanford Sentiment Treebank

Method	Fine-grained	Binary
RAE (Socher et al., 2013)	43.2	82.4
MV-RNN (Socher et al., 2013)	44.4	82.9
RNTN (Socher et al., 2013)	45.7	85.4
DCNN (Blunsom et al., 2014)	48.5	86.8
Paragraph-Vec (Le and Mikolov, 2014)	48.7	87.8
CNN-non-static (Kim, 2014)	48.0	87.2
CNN-multichannel (Kim, 2014)	47.4	**88.1**
DRNN (Irsoy and Cardie, 2014)	49.8	86.6
LSTM	45.8	86.7
Bidirectional LSTM	49.1	86.8
2-layer LSTM	47.5	85.5
2-layer Bidirectional LSTM	46.2	84.8
Constituency Tree LSTM (no tuning)	46.7	86.6
Constituency Tree LSTM	**50.6**	86.9

of word vectors

Semantic Similarity

- Better than binary paraphrase classification!
- Dataset from a competition:
  SemEval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness [and textual entailment]

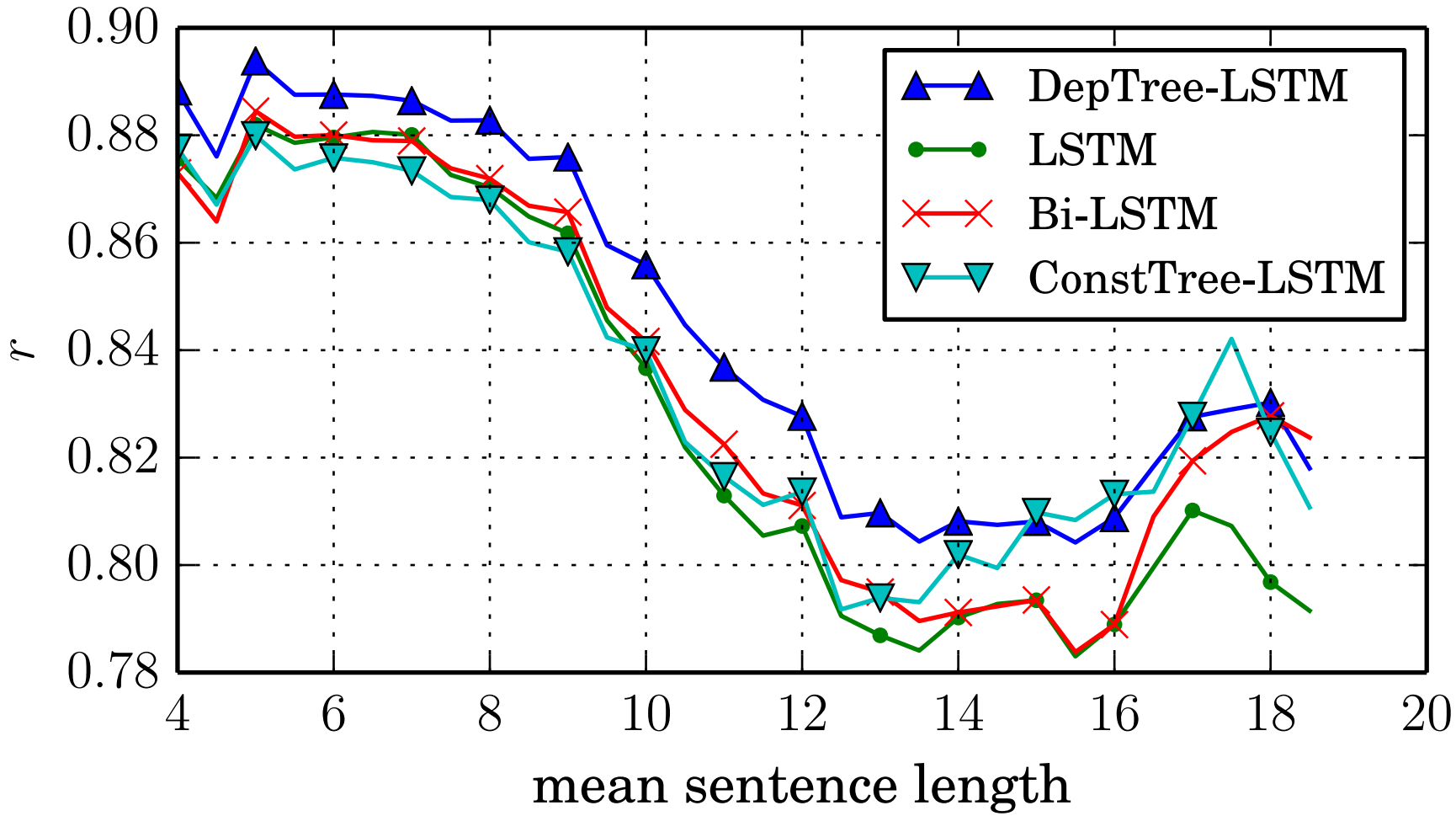
Relatedness score	Example
1.6	A: "*A man is jumping into an empty pool*" B: "*There is no biker jumping in the air*"
2.9	A: "*Two children are lying in the snow and are making snow angels*" B: "*Two angels are making snow on the lying children*"
3.6	A: "*The young boys are playing outdoors and the man is smiling nearby*" B: "*There is no boy playing outdoors and there is no man smiling*"
4.9	A: "*A person in a black jacket is doing tricks on a motorbike*" B: "*A man in a black jacket is doing tricks on a motorbike*"

Semantic Similarity Results (correlation and MSE)

Pearson's r, Spearman's ρ

Method	r	ρ	**MSE**
Mean vectors	0.8046	0.7294	0.3595
DT-RNN (Socher et al., 2014)	0.7863	0.7305	0.3983
SDT-RNN (Socher et al., 2014)	0.7886	0.7280	0.3859
Illinois-LH (Lai and Hockenmaier, 2014)	0.7993	0.7538	0.3692
UNAL-NLP (Jimenez et al., 2014)	0.8070	0.7489	0.3550
Meaning Factory (Bjerva et al., 2014)	0.8268	0.7721	0.3224
ECNU (Zhao et al., 2014)	0.8414	–	–
LSTM	0.8477	0.7921	0.2949
Bidirectional LSTM	0.8522	0.7952	0.2850
2-layer LSTM	0.8411	0.7849	0.2980
2-layer Bidirectional LSTM	0.8488	0.7926	0.2893
Constituency Tree LSTM	0.8491	0.7873	0.2852
Dependency Tree LSTM	**0.8627**	**0.8032**	**0.2635**

Semantic Similarity Results, Pearson Correlation



Next week: Review Session and Midterm

- Go over materials
- Visit office hours for PSet solutions
- Derive

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

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