Recursive neural networks for semantic interpretation

Sam Bowman

Department of Linguistics and NLP Group

Stanford University

with help from Chris Manning, Chris Potts, Richard Socher, Jeffrey Pennington, J.T. Chipman

Recent progress on deep learning

Neural network models are starting to seem pretty good at capturing aspects of meaning.

From Stanford NLP alone:

- Sentiment (EMNLP ‘11, EMNLP ‘12, EMNLP ‘13)
- Paraphrase detection (NIPS ‘11)
- Knowledge base completion (NIPS ‘13, ICLR ‘13)
- Word–word translation (EMNLP ‘13)
- Parse evaluation (NIPS ‘10, NAACL ‘12, ACL ‘13)
- Image labelling (ICLR ‘13)

Recent progress on deep learning

Wired, Jan 2014:

> Where will this next generation of researchers take the deep learning movement? The big potential lies in deciphering the words we post to the web — the status updates and the tweets and instant messages and the comments — and there's enough of that to keep companies like Facebook, Google, and Yahoo busy for an awfully long time.

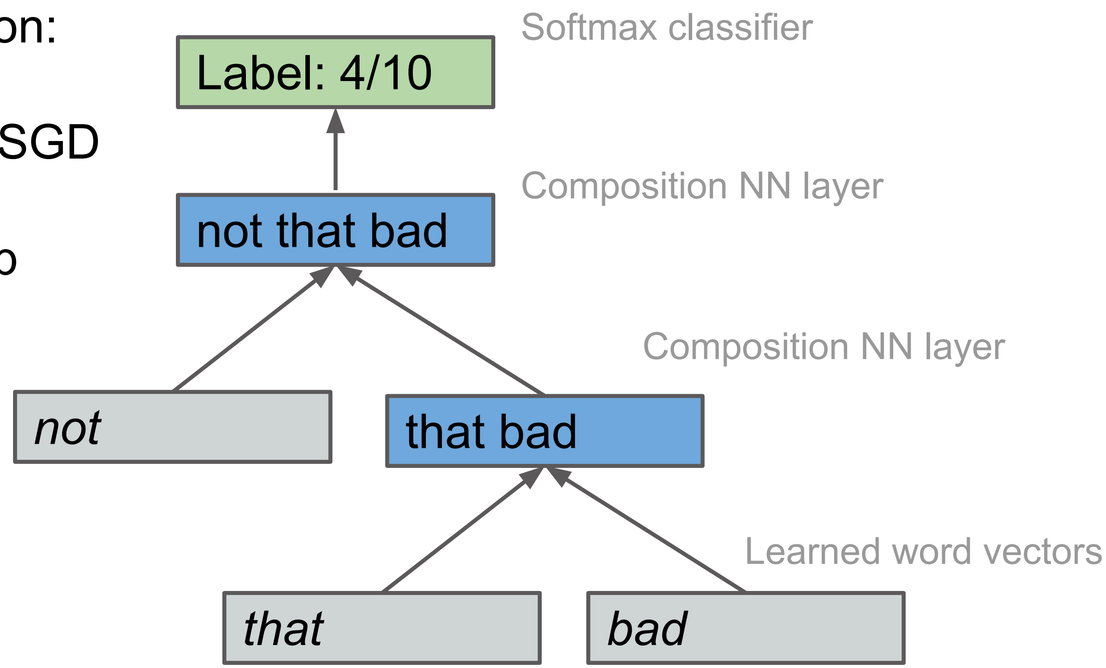
Today

Can these techniques learn models for general purpose NLU?

- **Survey: Deep learning models for NLU**
- Experiment: Can RNTNs learn to reason with quantifiers (in an ideal world)?
- Experiment: Can RNTNs learn the natural logic *join* operator?
- Experiment: How do these models do on a challenge dataset?

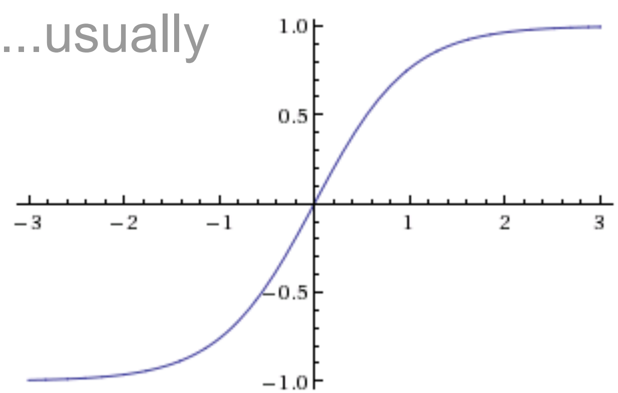
Recursive neural networks for text

- Words and constituents are ~50 dimensional vectors.
- RNN composition function:
  $y = f(Mx + b)$
- Optimize with AdaGrad SGD or L-BFGS
- Gradients from backprop (through structure)



$f(x) = \tanh(x)$

...usually

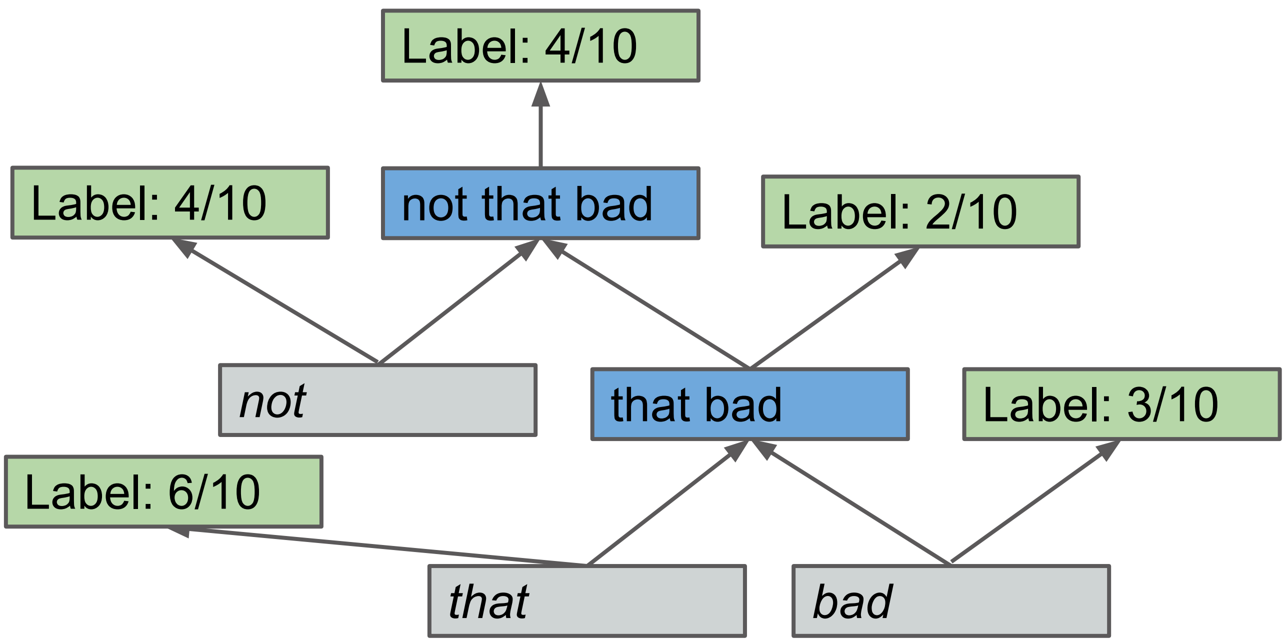


Socher et al. 2011

Recursive neural networks for text

Supervision for everyone!

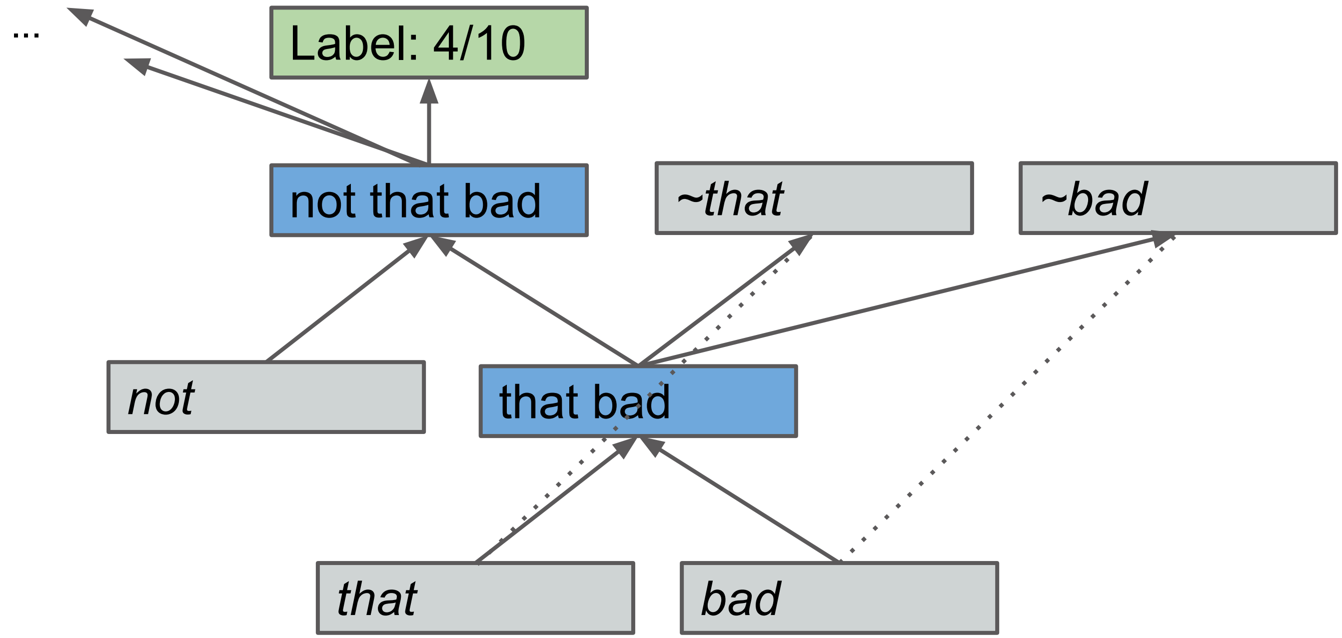
- ~10k sentences
- ~200k sentiment labels from mechanical Turk



Socher et al. 2013

Recursive neural networks for text

- Recursive autoencoder
- Two objectives: Classification and reconstruction

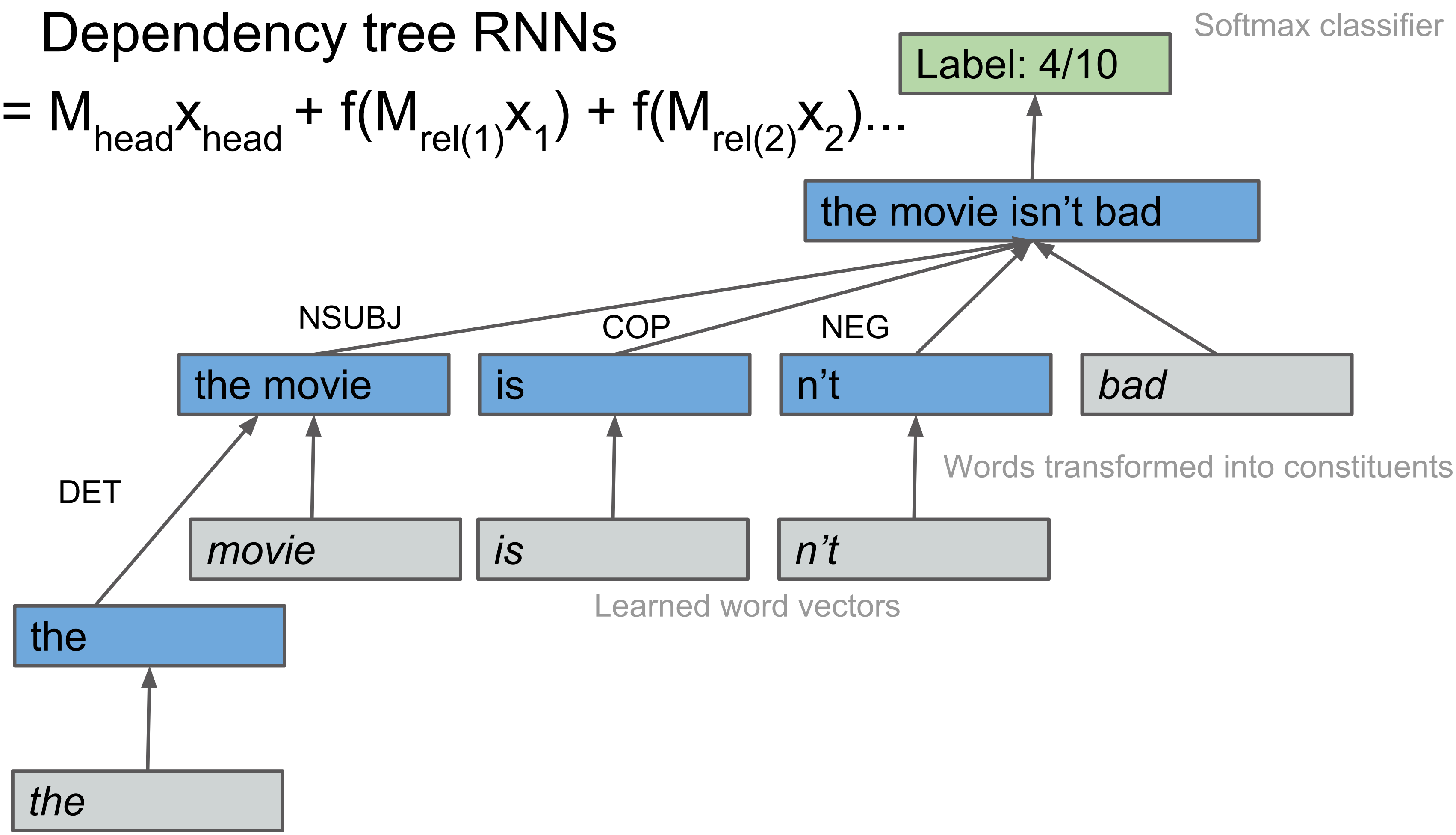


Socher et al. 2011

Recursive neural networks for text

- Dependency tree RNNs

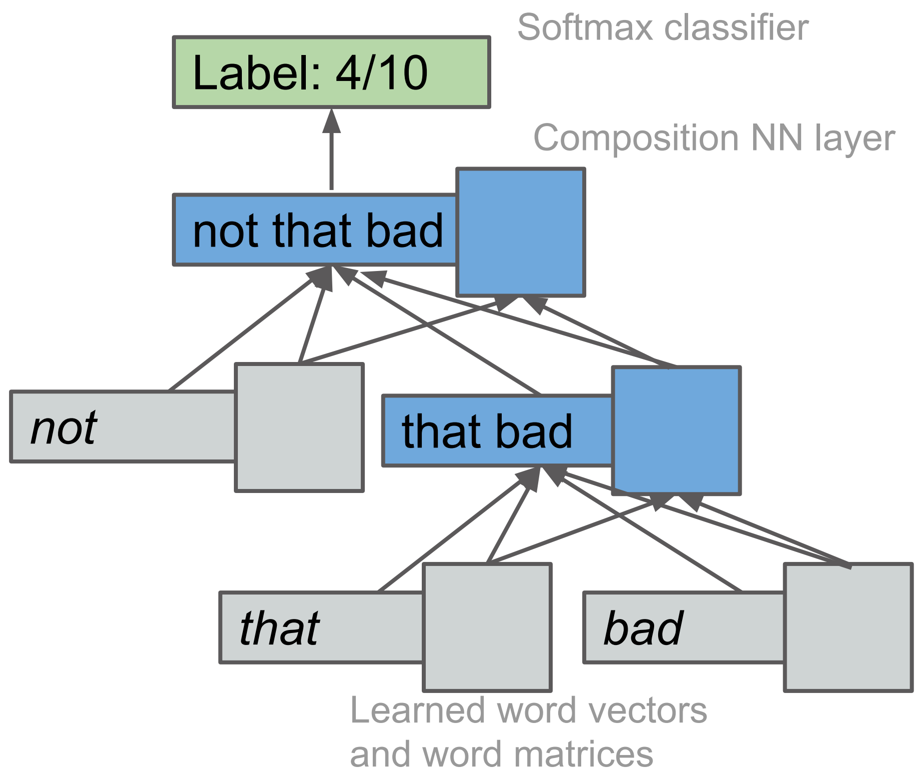
$$y = M_{head}x_{head} + f(M_{rel(1)}x_1) + f(M_{rel(2)}x_2)\ldots$$



Socher et al. 2014

Recursive neural networks for text

- Matrix-vector RNN composition functions:
  $y = f(M_v[Ba; Ab])$
  $Y = M_m[A; B]$

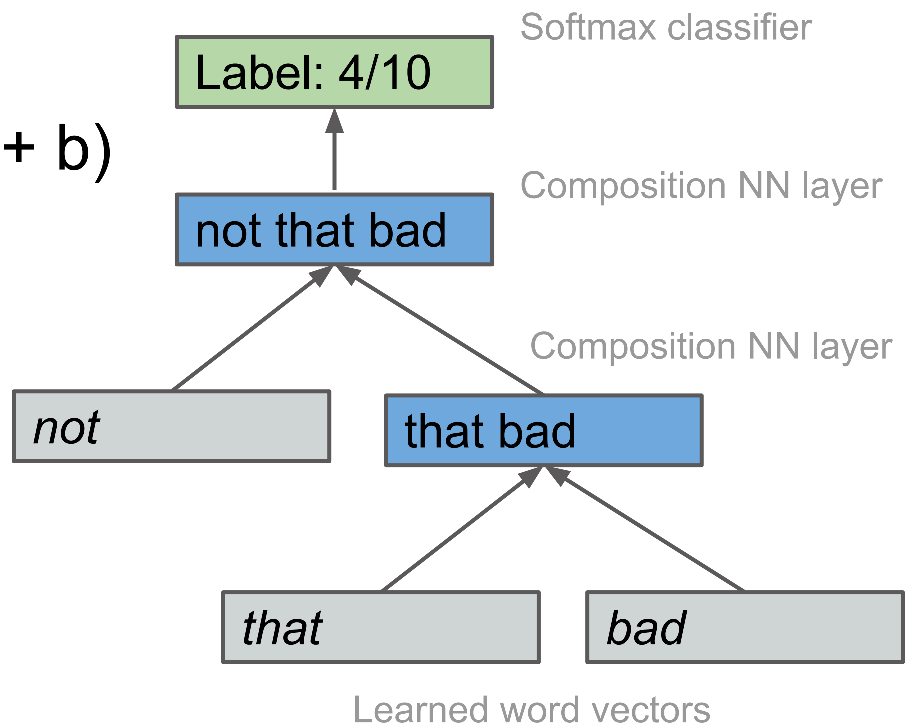


Socher et al. 2012

Recursive neural networks for text

- Recursive neural tensor network composition function:

$$y = f(x_1 M^{[1 \ldots N]} x_2 + Mx + b)$$



Chen et al. 2013, Socher et al. 2013

Recursive neural networks for text

And more:

- Convolutional RNNs (Kalchbrenner, Grefenstette, and Blunsom 2014)
- Bilingual objectives (Hermann and Blunsom 2014)

...

And this isn’t even considering model structures for language modeling or speech recognition...

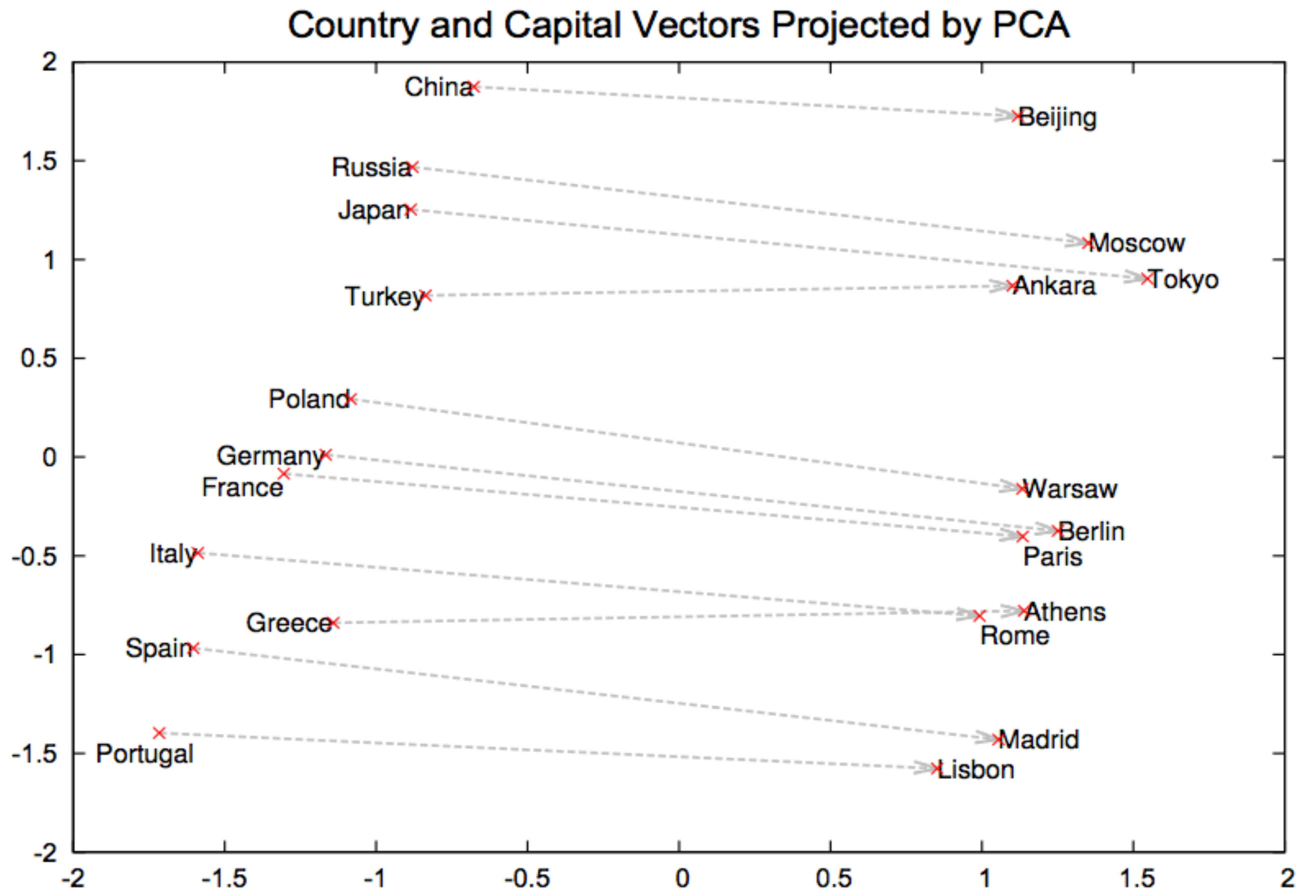
Today

Can these techniques learn models for general purpose NLU?

- Survey: Deep learning models for NLU
- **Experiment: Can RNTNs learn to reason with quantifiers (in an ideal world)?**
- Experiment: Can RNTNs learn the natural logic *join* operator?
- Experiment: How do these models do on a challenge dataset?

The problem

Mikolov et al. 2013, NIPS



The problem

The Mikolov et al. result:

- Paris - France + Spain = Madrid
- Paris - France + USA = ?
- most - some + all = ?
- not = ?

The problem

- Relatively little work to date on the expressive power of this kind of model.
- The goal of the project:

  Can the representation learning systems used in practice capture every aspect of meaning that formal semantics says language users need?

- This talk:

  Can RNNs learn to accurately reason with quantification and monotonicity?

Strict unambiguous NLI

- Hard to test on `world` ↔ *sentence*. (Why?)
- What about *sentence* ↔ *sentence*?
- Natural language inference (NLI):

  Doing logical inference where the logical formulae are represented using natural language.

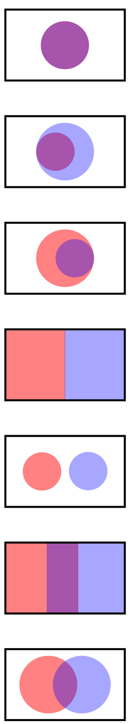
  (as formalized for NLP here by MacCartney, '09)

- Framed as classification task:
  - All dogs bark and Fido is a dog. $\sqsubset$ Fido barks.
  - No dog barks. ≡ All dogs don't bark.
  - No dog barks. **?** Some dog barks.

Strict unambiguous NLI

- MacCartney's seven possible relations between phrases/sentences:

Slide from Bill MacCartney



$x \equiv y$	equivalence	*couch* $\equiv$ *sofa*
$x \sqsubset y$	forward entailment (strict)	*crow* $\sqsubset$ *bird*
$x \sqsupset y$	reverse entailment (strict)	*European* $\sqsupset$ *French*
$x \wedge y$	negation (exhaustive exclusion)	*human* $\wedge$ *nonhuman*
$x \mid y$	alternation (non-exhaustive exclusion)	*cat* $\mid$ *dog*
$x \smile y$	cover (exhaustive non-exclusion)	*animal* $\smile$ *nonhuman*
$x \# y$	independence	*hungry* # *hippo*

Monotonicity (a quick reminder)

- A way of using lexical knowledge to reason about sentences.
- Given: black dogs $\sqsubset$ dogs, dogs $\sqsubset$ animals
  - Upward monotone:
    - *some dogs bark* $\sqsubset$ *some animals bark*
  - Downward monotone:
    - *all dogs bark* $\sqsubset$ *all black dogs bark*
  - *Non-monotone:*
    - *most dogs bark # most animals bark*
    - *most dogs bark # most black dogs bark*

Strict unambiguous NLI

Strip away everything *else* that makes natural language hard:

- Small, unambiguous vocabulary
- No morphology (no tense, no plurals, no agreement..)
- No pronouns/references to context
- Unlabeled constituency parses are given in data

The setup

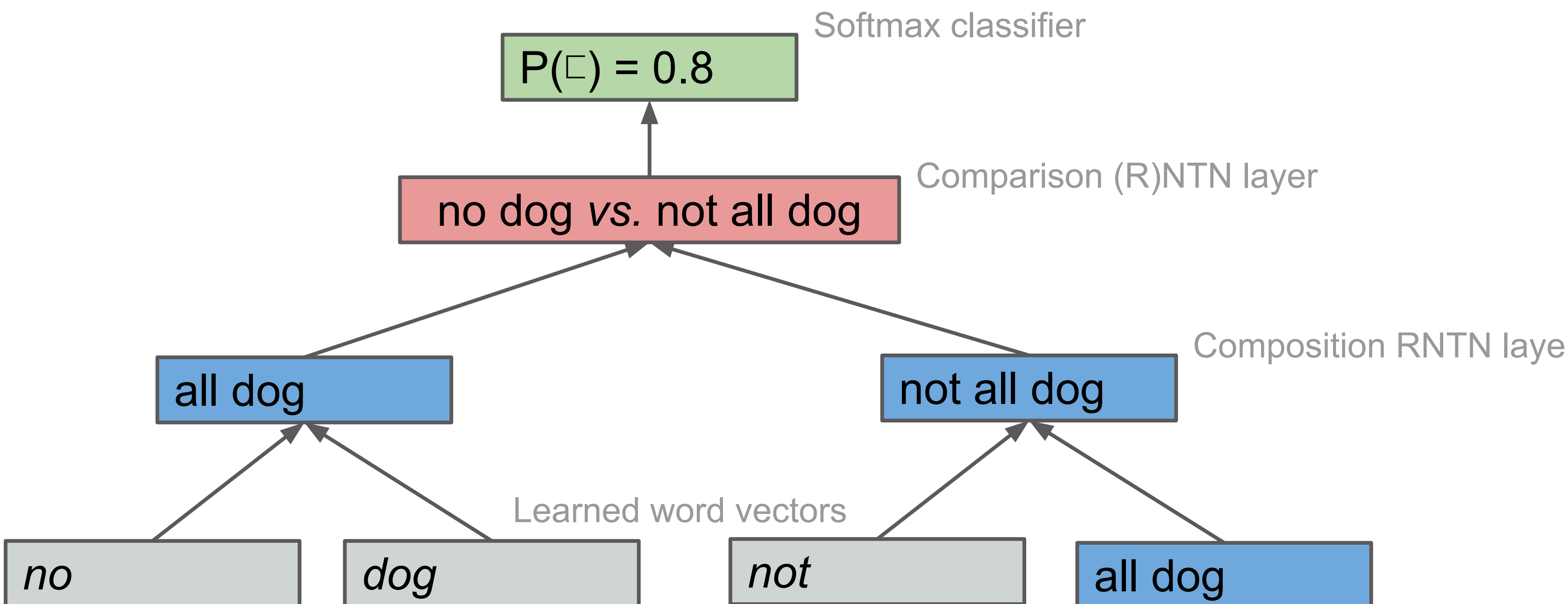
- Small (~50 word) vocabulary
  - Three basic types:
    - Quantifiers: *some*, *all*, *no*, *most*, *two*, *three*
    - Predicates: *dog, cat, animal, live, European, …*
    - Negation: *not*
- Handmade dataset, 12k sentence pairs, grouped into templates.
- All sentences of the form *QPP*, with optional negation on each predicate:

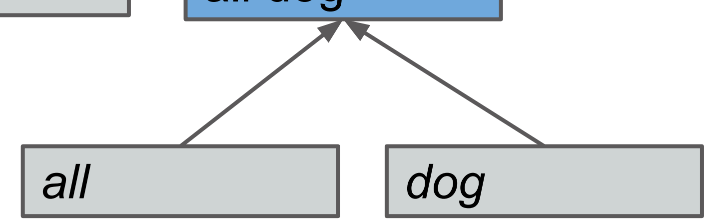
  ((some x*) bark) # ((some* x*) (not bark))*

  ((some dog) bark) # ((some dog) (not bark))

  ((most (not dog)) European) ⊐ *((most (not dog)) French)*

The model: an RNTN for NLI





- Layers are parameterized with third-order tensors, after Chen et al. ‘13
- Parameters are shared between copies of the composition layer
- Input word vectors are initialized randomly and learned.

Five experiments

- *All-in:* train and test on all data. ⇒ 100%
- *All-split:* train on 85% of each pattern, test on rest.
  ⇒ 100%

```
(most dog) bark | (no dog) alive
(all cat) French ⊐ (some cat) European

(most dog) French | (no dog) European
```

Five experiments

- *One-set-out:* hold out one pattern for testing only, split remaining data 85/15.
  - `(most` *x*`) European | (no` *x*`) European`
- *One-subclass-out:* hold out one set of patterns for testing only, split remaining data 85/15.
  - `(most` *x*`)` *y* `| (no` *x*`)` *y*
- *One-pair-out:* hold out one every pattern with a given pair of quantifiers for testing only, split rest.
  - `(most (not` *x*`))` *y* `# (no` *x*`)` *z*`...`

Pilot results

Target dataset	Data evaluated	SET-OUT	SUBCL.-OUT	PAIR-OUT
(most x) bark $\mid$ *(no x)* bark	target dataset only	100%	100%	93.6%
	all held out datasets	(100%)	36.8%	78.8%
	all test data	99.8%	95.9%	93.8%
(two x) bark # (all x) bark	target dataset only	0%	100%	94.7%
	all held out datasets	(0%)	100%	62.7%
	all test data	97.5%	99.3%	93.0%
(some x) bark ^ *(no x) bark*	target dataset only	0%	0%	0%
	all held out datasets	(0%)	0%	25.2%
	all test data	97.7%	94.0%	85.5%

MacCartney's join:

$(\mathit{most}\ x)\ y \sqsubset (\mathit{some}\ x)\ y\ ,\ (\mathit{some}\ x)\ y\ \wedge\ (\mathit{no}\ x)\ y$

$\models\ (\mathit{most}\ x)\ y \mid (\mathit{no}\ x)\ y$

$(\mathit{some}\ x)\ y \sqsupset (\mathit{most}\ x)\ y\ ,\ (\mathit{most}\ x)\ y \mid (\mathit{no}\ x)\ y$

$\models\ (\mathit{some}\ x)\ y\ \{\sqsupset \wedge \mid \# \smile\}\ (\mathit{no}\ x)\ y$

Today

Can these techniques learn models for general purpose NLU?

- Survey: Deep learning models for NLU
- Experiment: Can RNTNs learn to reason with quantifiers (in an ideal world)?
- **Experiment: Can RNTNs learn the natural logic *join* operator?**
- Experiment: How do these models do on a challenge dataset?

Extra experiments: MacC's Join

$\bowtie$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\equiv$	$\equiv$	$\sqsubset$	$\sqsupset$	$\wedge$	$\mid$	$\smile$	$\#$
$\sqsubset$	$\sqsubset$	$\sqsubset$	$\equiv\sqsubset\sqsupset\mid\#$	$\mid$	$\mid$	$\sqsubset\wedge\mid\smile\#$	$\sqsubset\mid\#$
$\sqsupset$	$\sqsupset$	$\equiv\sqsubset\sqsupset\smile\#$	$\sqsupset$	$\smile$	$\sqsupset\wedge\mid\smile\#$	$\smile$	$\sqsupset\smile\#$
$\wedge$	$\wedge$	$\smile$	$\mid$	$\equiv$	$\sqsupset$	$\sqsubset$	$\#$
$\mid$	$\mid$	$\sqsubset\wedge\mid\smile\#$	$\mid$	$\sqsubset$	$\equiv\sqsubset\sqsupset\mid\#$	$\sqsubset$	$\sqsubset\mid\#$
$\smile$	$\smile$	$\smile$	$\sqsupset\wedge\mid\smile\#$	$\sqsupset$	$\sqsupset$	$\equiv\sqsubset\sqsupset\smile\#$	$\sqsupset\smile\#$
$\#$	$\#$	$\sqsubset\smile\#$	$\sqsupset\mid\#$	$\#$	$\sqsupset\mid\#$	$\sqsubset\smile\#$	$\equiv\sqsubset\sqsupset\wedge\mid\smile\#$

MacCartney's join table: *aRb & bR'c ⇒ a{join(R,R')}c*

Cells that contain # represent uncertain results and can be approximated by just #.

Extra experiments: Lattices with join

EXTRACTED RELATIONS:

$b \equiv b$

$b \smile c$

$b \smile d$

$b \sqsupset e$

$c \smile d$

$c \sqsupset e$

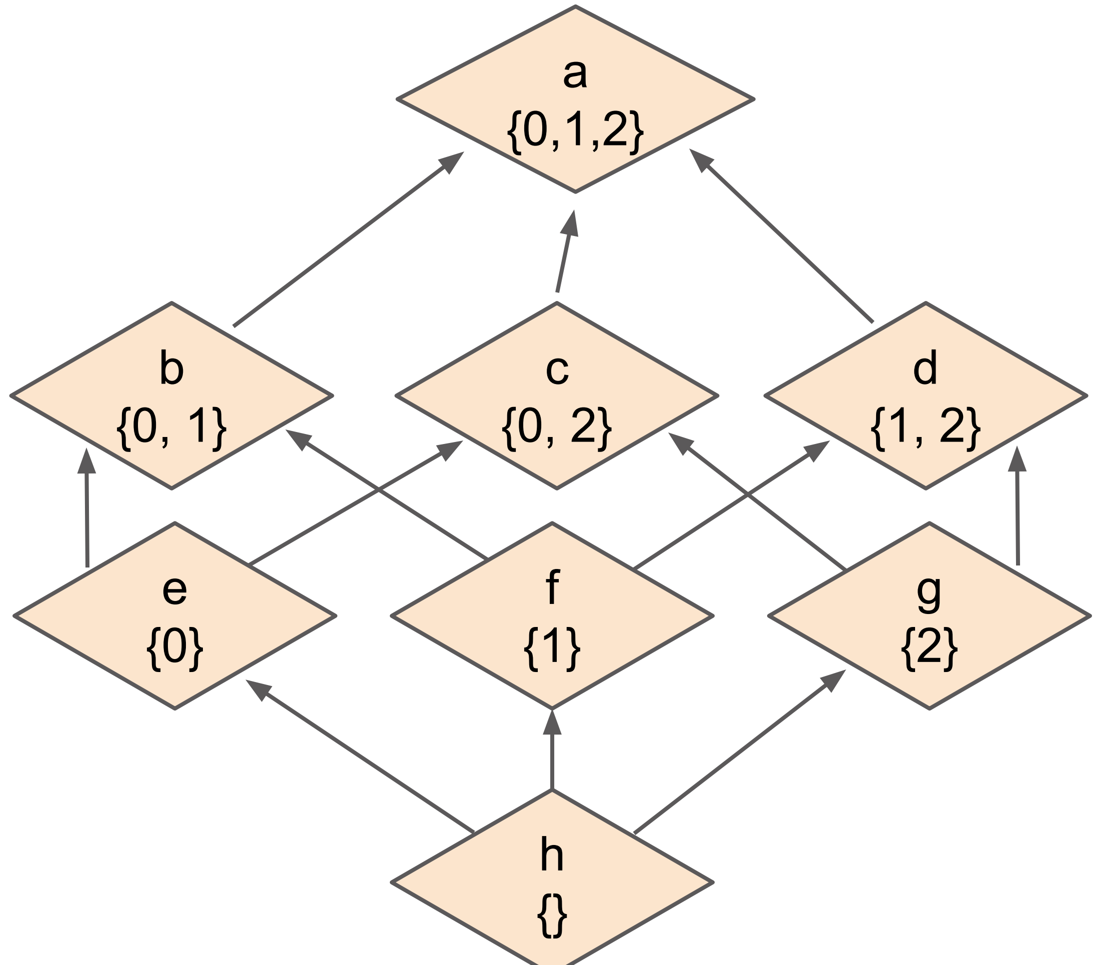
$c \wedge f$

$c \sqsupset g$

$e \sqsubset b$

$e \sqsubset c$

...



Extra experiments: Lattices with join

EXTRACTED RELATIONS:

$b \equiv b$

$\mathrm{b} \smile \mathrm{c}$

$b \smile d$

$b \sqsupset e$

$\mathrm{c} \smile \mathrm{d}$

$\mathrm{c} \sqsupset \mathrm{e}$

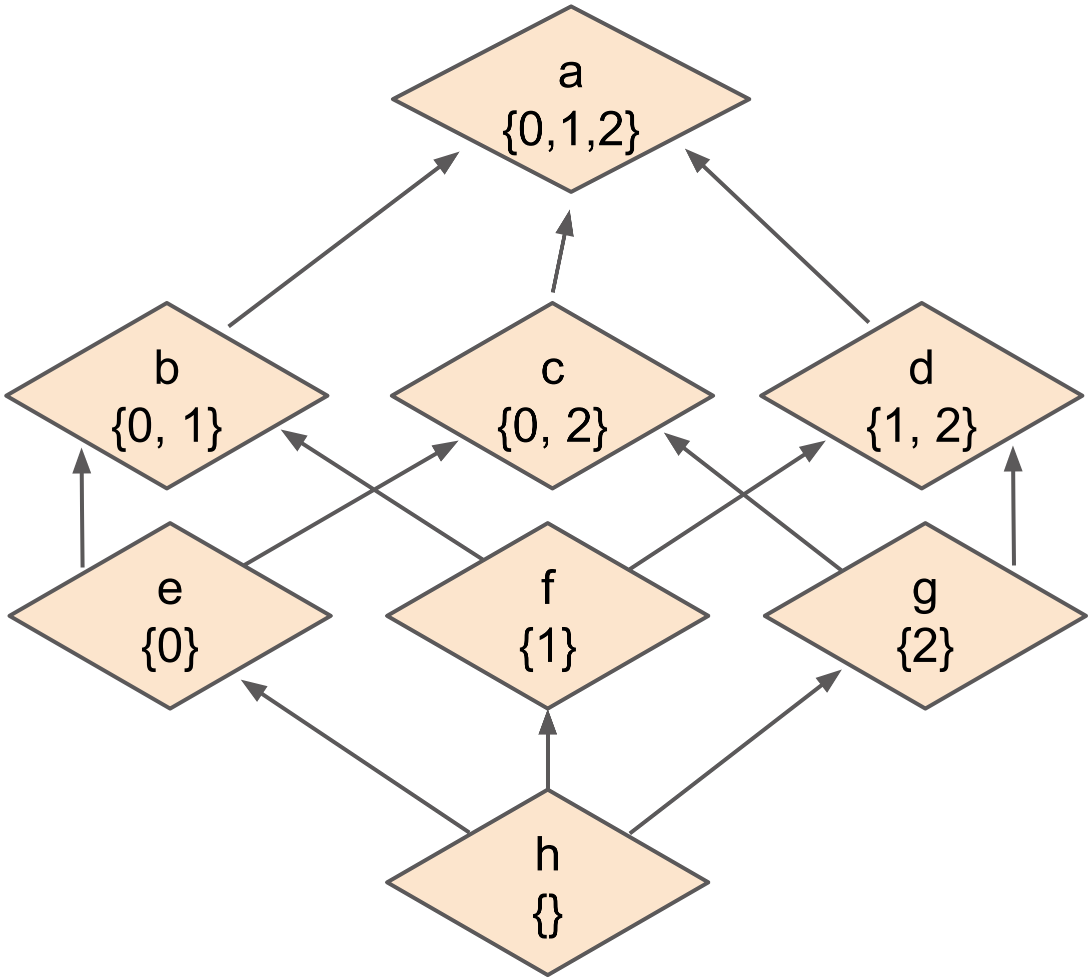
$c \wedge f$

$\mathrm{c} \sqsupset \mathrm{g}$

$e \sqsubset b$

$\mathrm{e} \sqsubset \mathrm{c}$

...



Extra experiments: Lattices with join

TRAIN:

b ⌣ c

c ⌣ d

c ⊐ e

c ⊐ g

e ⊏ c

...

TEST:

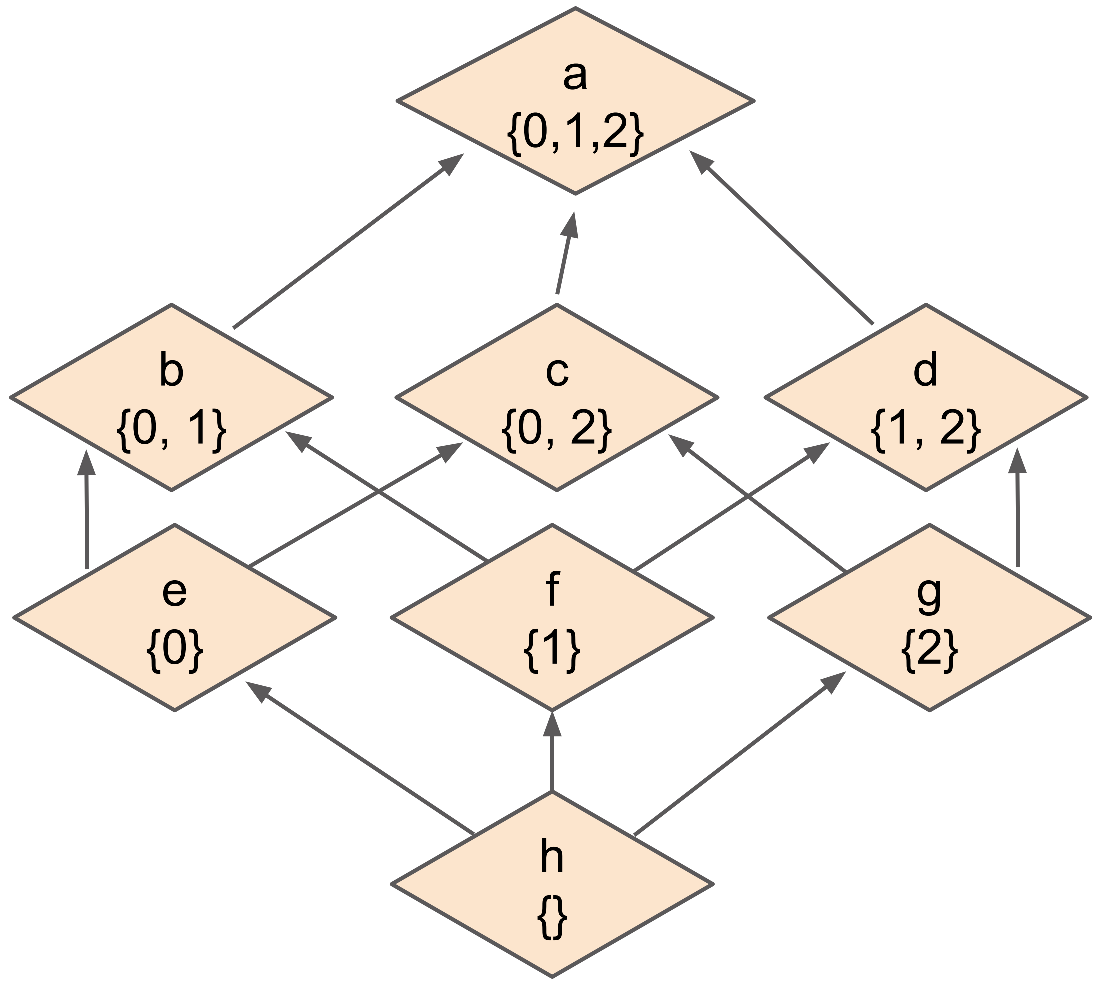
b ≡ b

b ⌣ d

b ⊐ e

c ^ f

e ⊏ b



Extra experiments: Lattices with join

TRAIN:

$b \smile c$

$c \smile d$

$c \sqsupset e$

$c \sqsupset g$

$e \sqsubset c$

...

TEST:

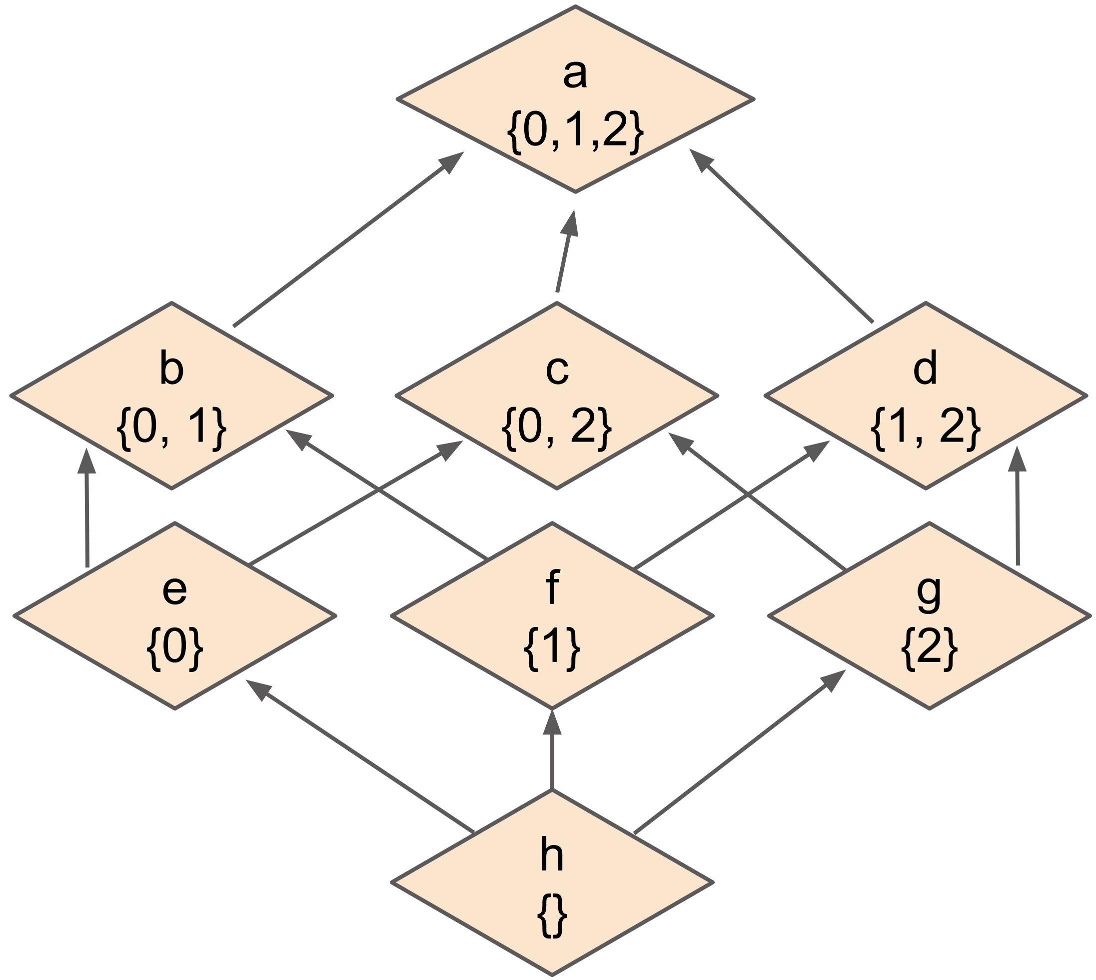
$b \equiv b$

$b \smile d$

$\cancel{b \sqsupset e}$

$\cancel{c \wedge f}$

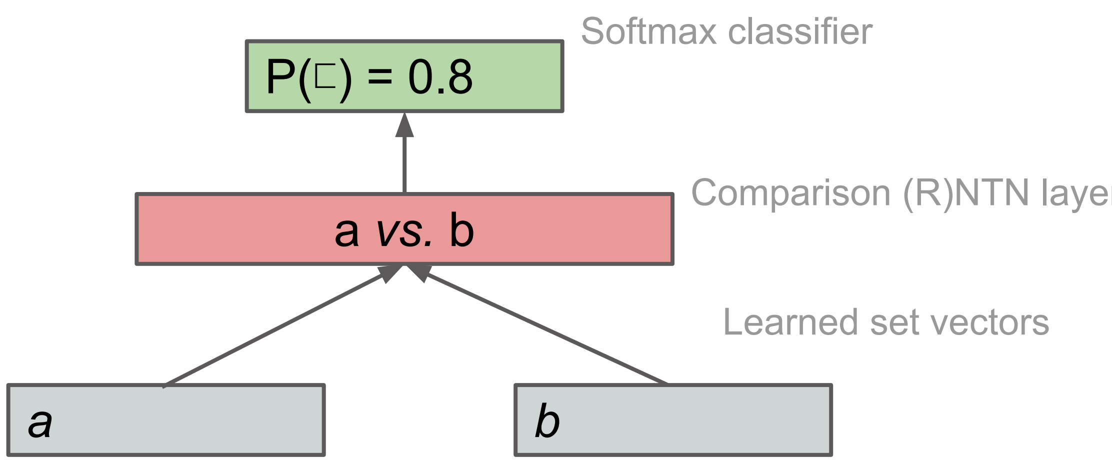
$e \sqsubset b$



Extra experiments: Lattices with join

- Same model as in the monotonicity experiments above, but no composition/internal structure in the sentences.
- Lattice with 50 sets/nodes, 50% of data held out for testing.

  ⇒ 100% accuracy



Today

Can these techniques learn models for general purpose NLU?

- Survey: Deep learning models for NLU
- Experiment: Can RNTNs learn to reason with quantifiers (in an ideal world)?
- Experiment: Can RNTNs learn the natural logic *join* operator?
- **Experiment: How do these models do on a challenge dataset?**

SemEval SICK

- NLP challenge dataset:
  - 10,000 sentence pairs labeled:
    - {forward entailment, contradiction, neutral}
  - “Sentences involving compositional knowledge” challenge:
    - No idioms, no named entities, no anaphora, tense doesn’t matter.
    - Requires general knowledge about word meaning and hypernymy, but no factoid knowledge.

SemEval SICK data

CONTRADICTION:

The woman in a red costume is leaning against a brick wall and playing an instrument.

The woman in a red costume is not leaning against a brick wall and is not playing an instrument.

NEUTRAL:

The player is dunking the basketball into the net and a crowd is in background.

A man with a jersey is dunking the ball at a basketball game.

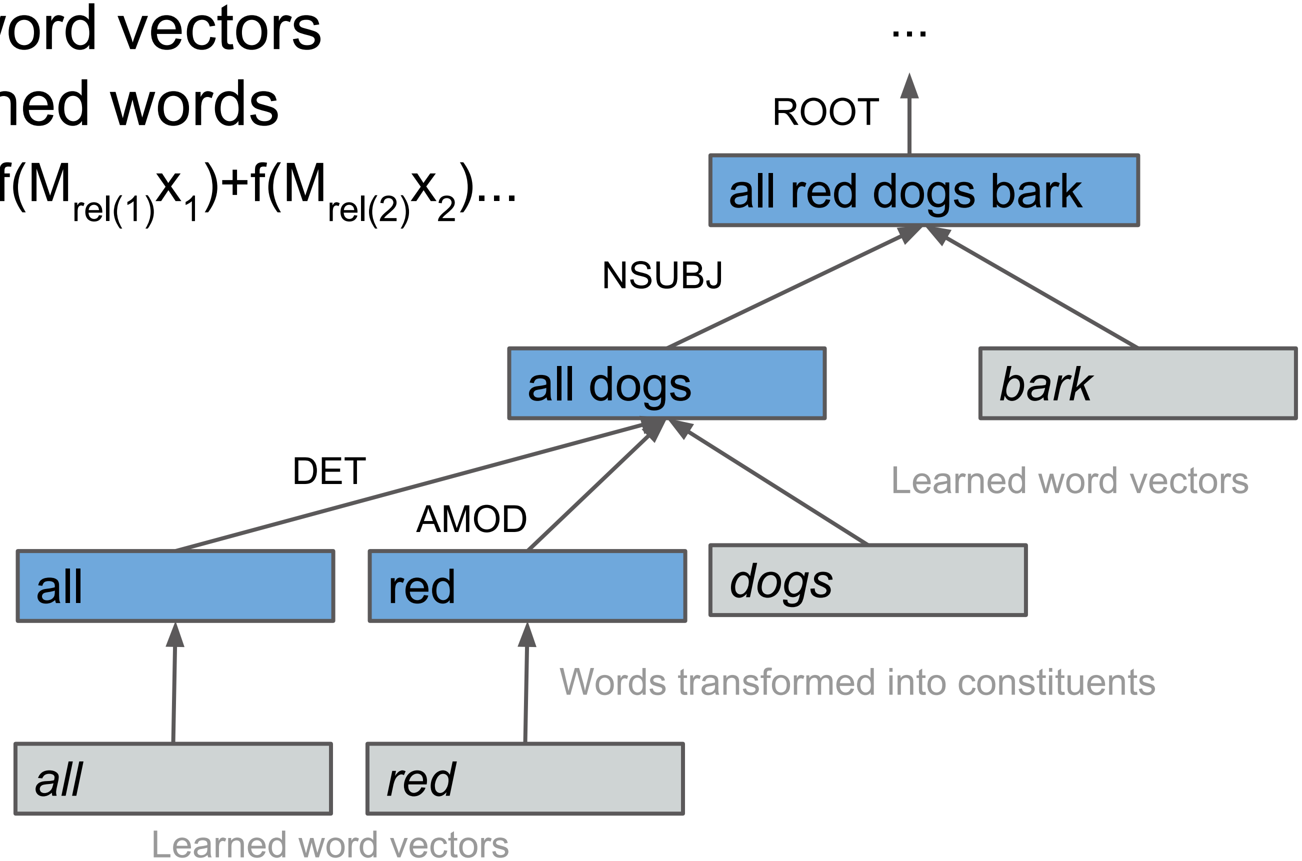
ENTAILMENT:

Four kids are doing backbends in the park

Four children are doing backbends in the park

SemEval SICK model

- Dependency tree RNNs
- Pretrained word vectors
- Partially-trained words
- $y = M_{head}x_{head}+f(M_{rel(1)}x_1)+f(M_{rel(2)}x_2)\ldots$



Results so far... eh?

- String inclusion baseline: 55.2%
- Most frequent class (Neutral): 56.4%
- Best dependency tree RNN: 74.5%
- Best SemEval result (UIllinois): 84.6%

But!

- No alignment or word sense disambiguation

Deep learning logistics

- There isn't any library yet that can do everything you'll need well.
  - But! Research code is available in MATLAB and Java
- Training monotonicity and SICK models: 4-18 hrs
- Lots of knobs to twiddle:
  - Stochastic optimization (AdaGrad/SGD) v. batch (L-BFGS)
  - Number of layers, dimensionality, L1 v. L2
  - Type of nonlinearity
  - Train/test split
  - DepTree RNNs: diagonal v. square matrices

...

Thanks!

Code is available for all three experiments.

sbowman@stanford.edu

Next steps

- Better formal characterizations of what it takes to learn to do inference
- Better formal characterizations of the structures that can be learned
- More types of network
- More semantic phenomena
- Test on natural language data